1. Za potrebe ovog sajta korišćene su sledeće tehnologije:

- HTML4
- HTML5
- CCS2
- CSS3
- JavaScript
- Jquery
- AJAX
- PHP
- JSON
- Laravel

2. Opis funkcionalnosti:

- Admin panel – brisanje, dodavnje ,ažuriranje korisnika, dodavanje, brisanje i izmena kategorija, dodavanje i brisanje i izmena sponzora, dodavanje, brisanje, ažuriranje serija, izveštaj aktivnosti korisnika
- Galerija – slike serija i sponzora(reklama)
- Registracija – pri završetku registracije korisniku se šalje mejl sa aktivacionim linkom i tek kada korisnik klikne na njega postaje aktivan korisnik, korišćen PHP mailer
- Logovanje – samo korisnici koji su se prethodno registrovali i aktivirali svoj nalog mogu se ulogovati
- Kontakt stranica – nakon što korisnik ispravno popuni sva polja i klikom na Pošalji, administratoru sajta se šalje mejl sa mejlom, imenom i prezimenom osobe koje želi da postavi pitanje
- Upotreba AJAX-a : logovanje, registracija, izmena korisnika, ,dodavanje kategorija, izmena kategorija, brisanje kategorija, kontakt stranica, klikom na neku seriju prikazuje se prošireni prikaz serija, prikaz komentara, pretraga serije po naslovu,

slanje komentara samo za prethodno registrovane korisnike, lajk i dislajk za komentar,

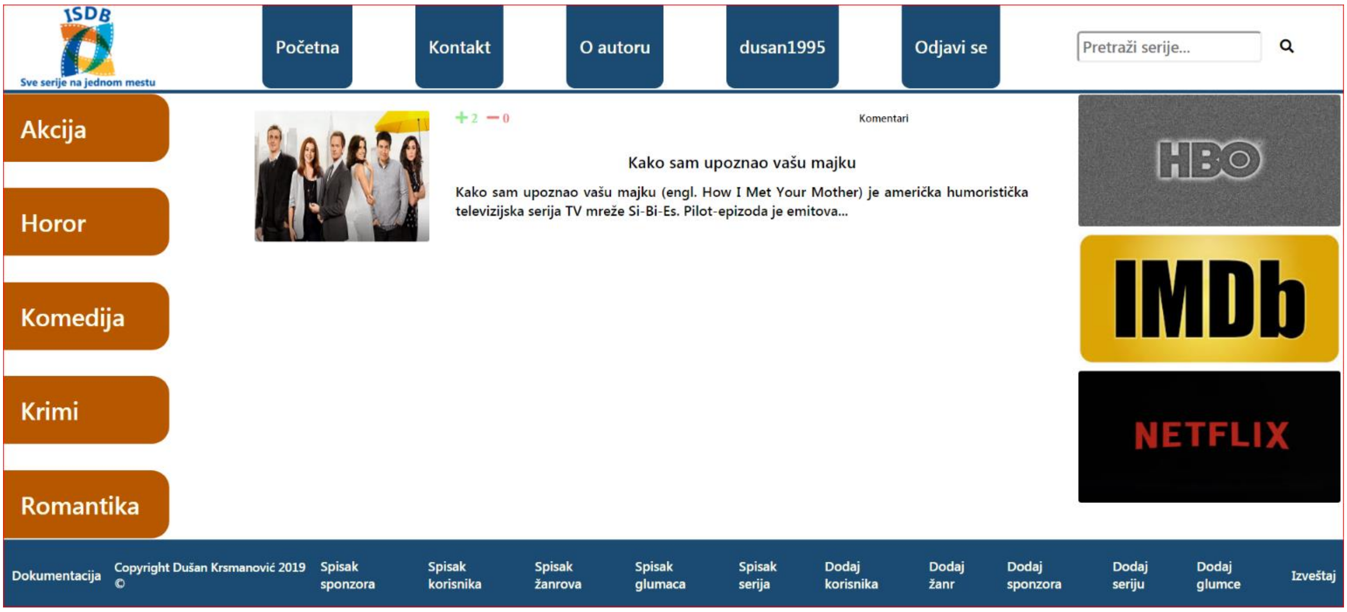
lajk i dislajk za seriju, dodavanje brisanje i izmena uloga za seriju,

- Dinamičko generisanje iz baze – Sadržaj koji je ključan (slike, serije, komentari...) se izvlači iz baze
- Anketa - Samo registrovani korisnici mogu ocenjivati komentare i serije, može glasati samo jednom za jedan komentar i seriju, ne može glasati za svoj komentar, onemogucavanje korisnika da dva puta glasa za isti komentar i seriju je uradjeno na nivou baze(unique za dve kolone)
- Korisnik može komentarisati serije, brisati sopstvene komentare
- Klikom na neku od potkategorija prikazuju se serije za odabranu kategoriju
- Na početnoj stranici se prikazuju prvih 5 najnovijih serija po godini izdanja

3.Stranice

korisnički deo sajta

pocetna

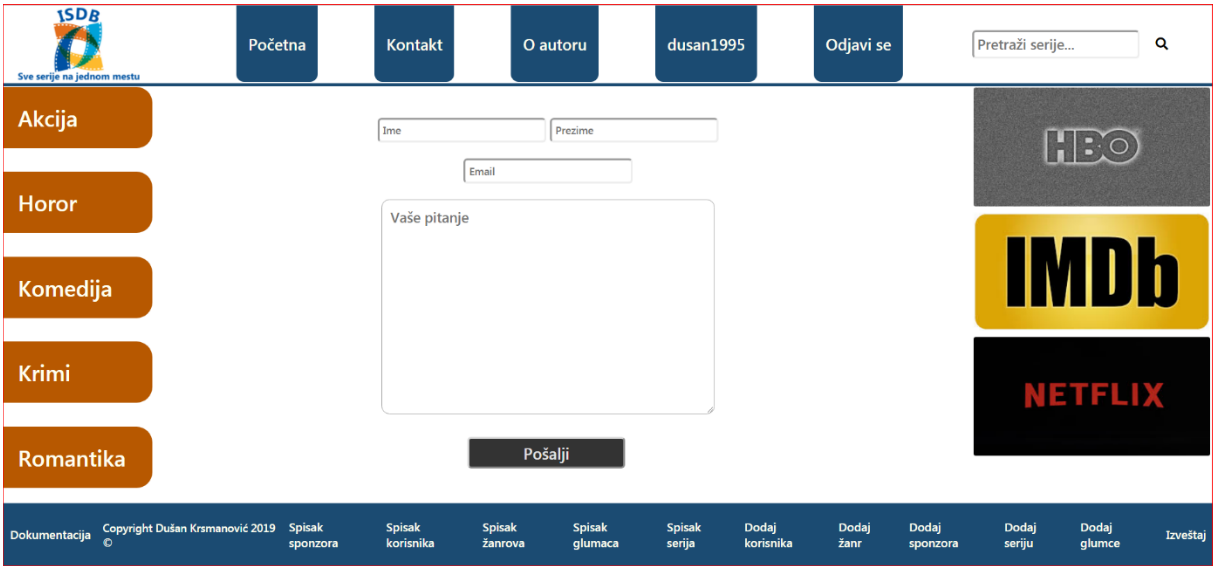


Kako sam upoznao vašu majku

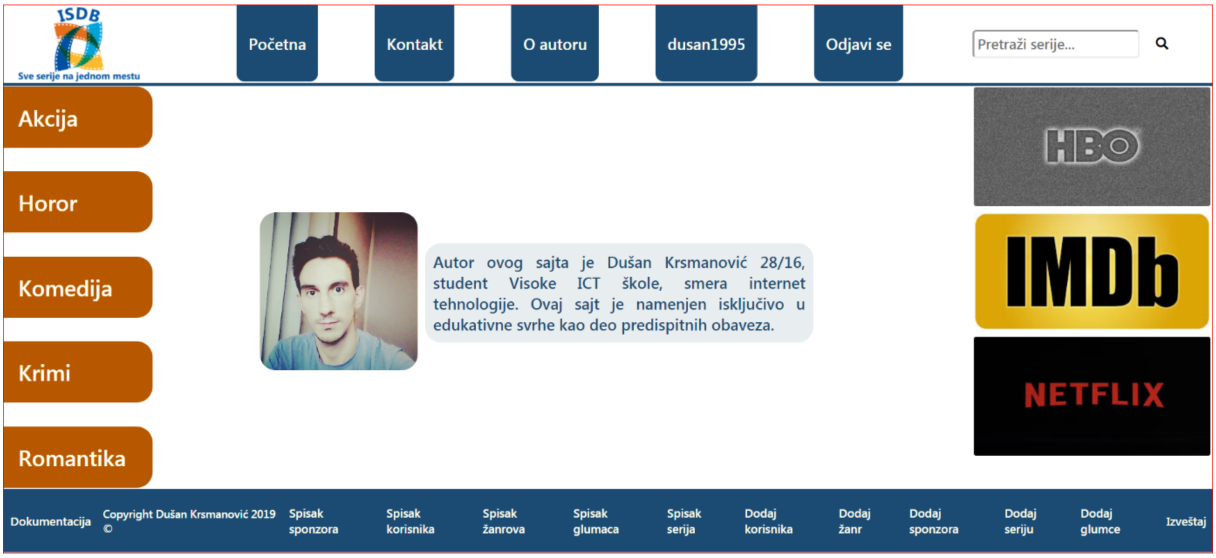
Kako sam upoznao vašu majku (engl. How I Met Your Mother) je američka humoristička televizijska serija TV mreže Si-Bi-Es. Pilot-epizoda je emitova...

Copyright Dušan Krsmanović 2019 ©

kontakt

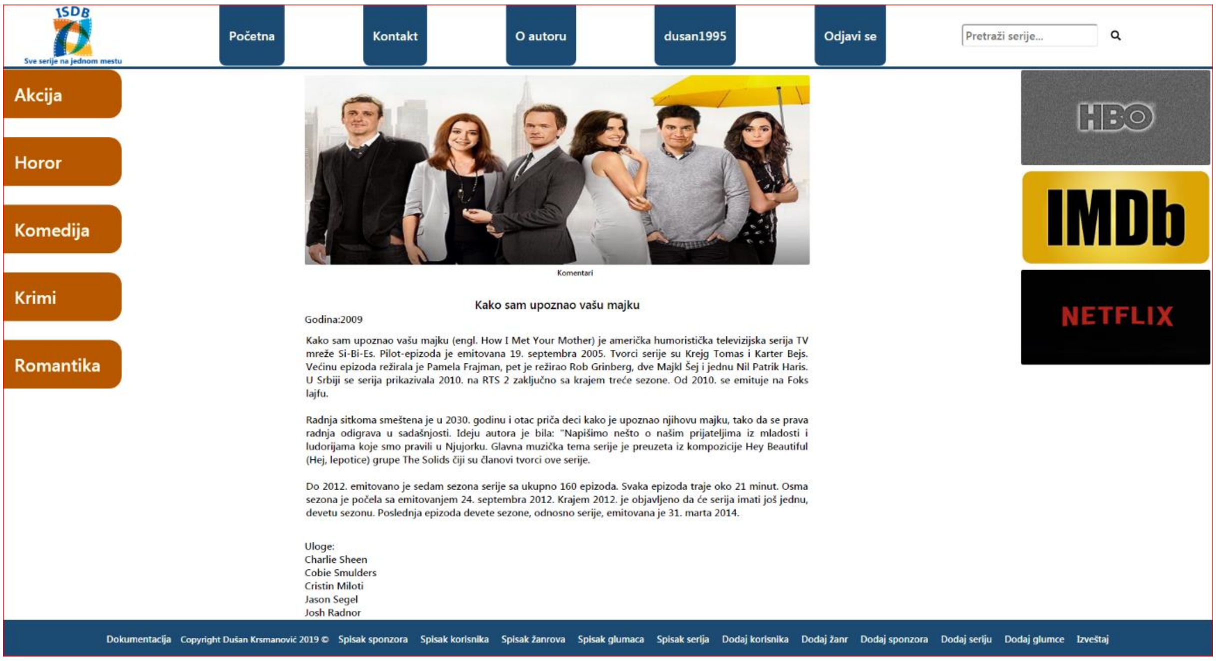


autor



Autor ovog sajta je Dušan Krsmanović 28/16, student Visoke ICT škole, smera internet tehnologije. Ovaj sajt je namenjen isključivo u edukativne svrhe kao deo predispitnih obaveza.

prošireni prikaz serije (dobijen ajaxom)



Kako sam upoznao vašu majku

Godina:2009

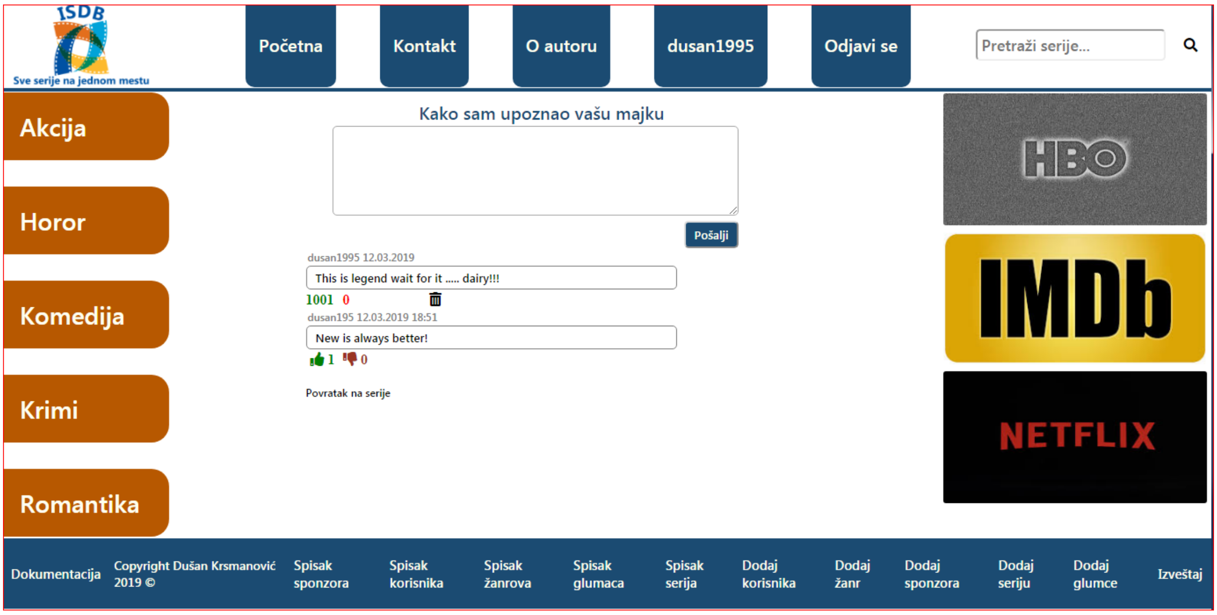
Kako sam upoznao vašu majku (engl. How I Met Your Mother) je američka humoristička televizijska serija TV mreže Si-Bi-Es. Pilot-epizoda je emitovana 19. septembra 2005. Tvorci serije su Krejg Tomas i Karter Bejs. Većinu epizoda režirala je Pamela Frajman, pet je režirao Rob Grinberg, dve Majkl Šej i jednu Nil Patrik Haris. U Srbiji se serija prikazivala 2010. na RTS 2 zaključno sa krajem treće sezone. Od 2010. se emituje na Foks lajfu.

Radnja sitkoma smeštena je u 2030. godinu i otac priča deci kako je upoznao njihovu majku, tako da se prava radnja odigrava u sadašnjosti. Ideju autora je bila: "Napišimo nešto o našim prijateljima iz mladosti i ludorijama koje smo pravili u Njujorku. Glavna muzička tema serije je preuzeta iz kompozicije Hey Beautiful (Hej, lepotice) grupe The Solids čiji su članovi tvorci ove serije.

Do 2012. emitovano je sedam sezona serije sa ukupno 160 epizoda. Svaka epizoda traje oko 21 minut. Osma sezona je počela sa emitovanjem 24. septembra 2012. Krajem 2012. je objavljeno da će serija imati još jednu, devetu sezonu. Poslednja epizoda devete sezone, odnosno serije, emitovana je 31. marta 2014.

Uloge:
Charlie Sheen
Cobie Smulders
Cristin Miloti
Jason Segel
Josh Radnor

prikaz komentara (dobijeno ajaxom)



Kako sam upoznao vašu majku

dusan1995 12.03.2019

This is legend wait for it ..... dairy!!!

1001 0

dusan195 12.03.2019 18:51

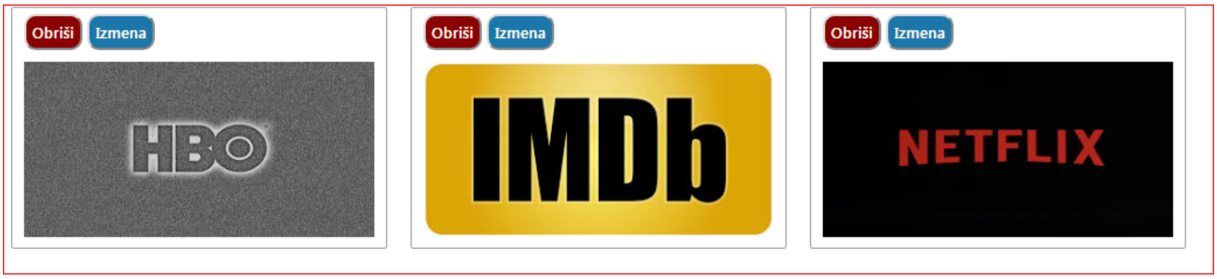
New is always better!

1 0

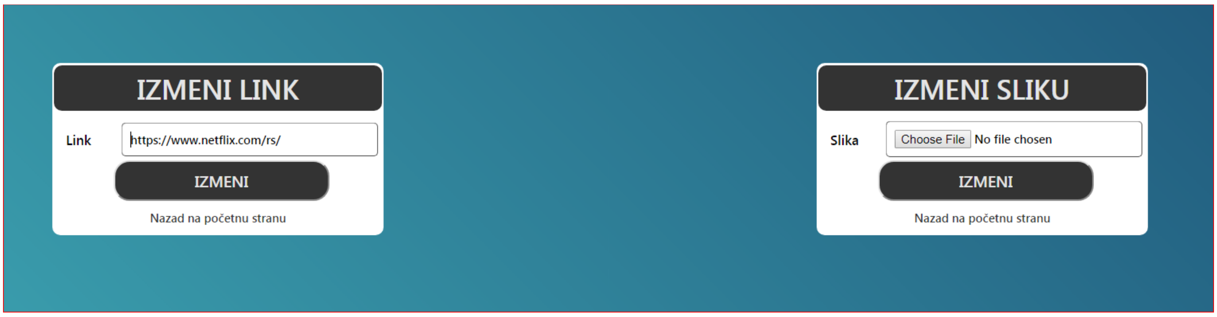
Povratak na serije

admin panel stranice

admin/sponzori/spisak



admin/sponzori/forma-izmena



IZMENI LINK

Link https://www.netflix.com/rs/

IZMENI

Nazad na početnu stranu

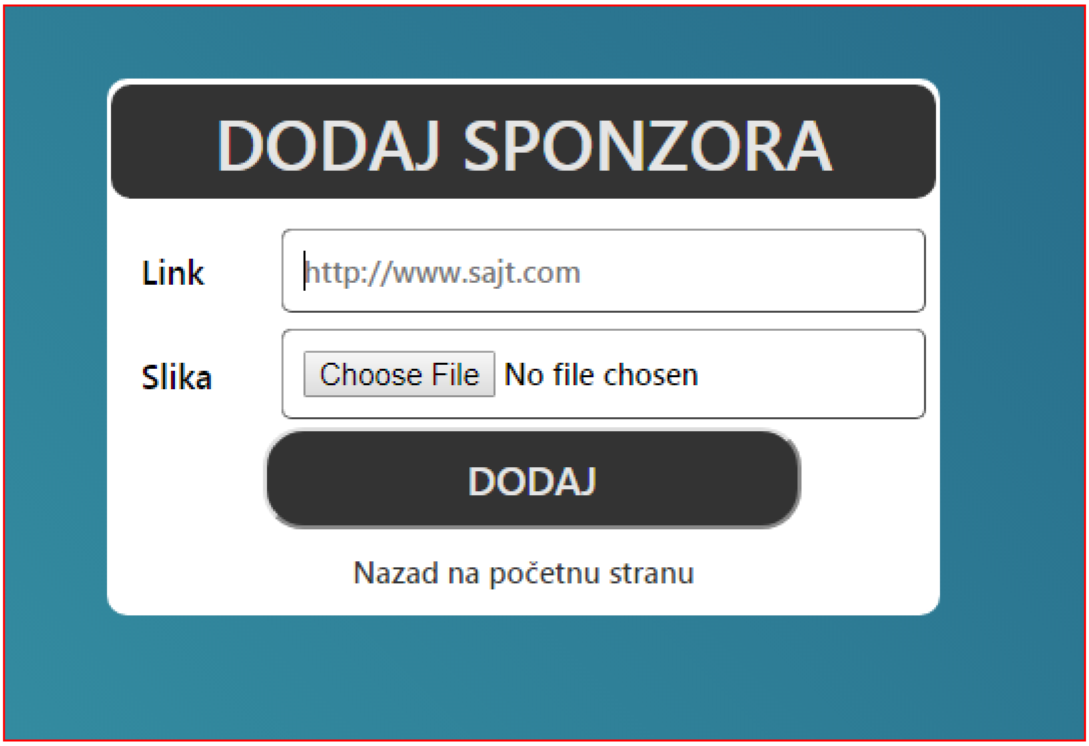
IZMENI SLIKU

Slika Choose File No file chosen

IZMENI

Nazad na početnu stranu

admin/sponzori/forma



DODAJ SPONZORA

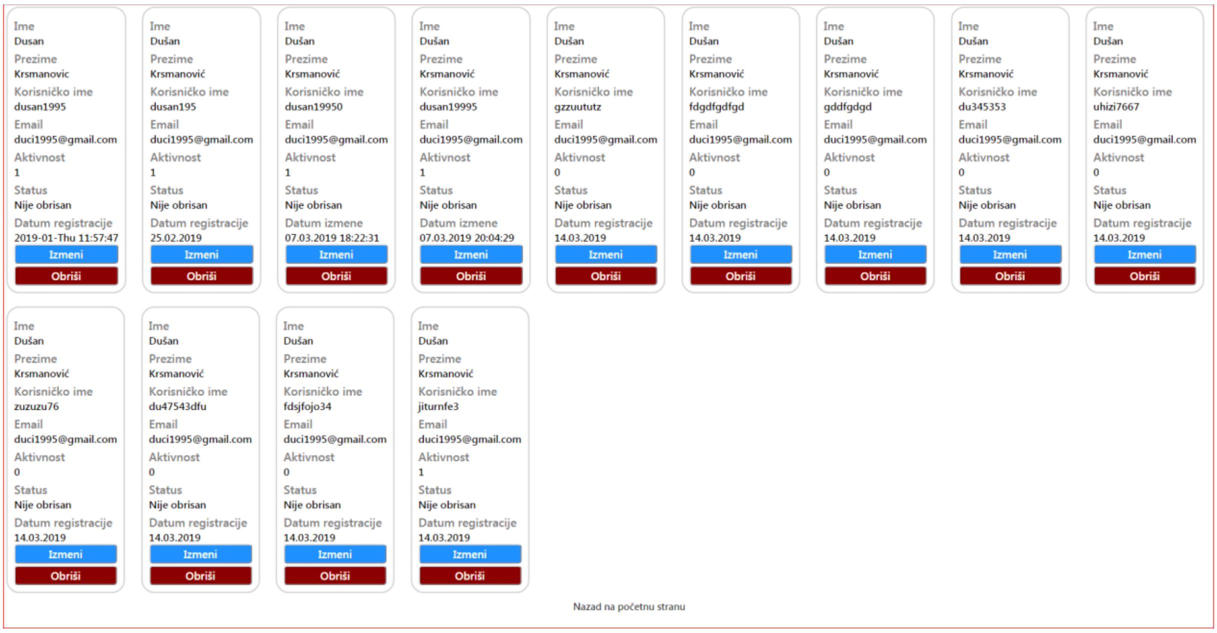
Link http://www.sajt.com

Slika Choose File No file chosen

DODAJ

Nazad na početnu stranu

admin/korisnici/spisak



Ime
Dusan
Prezime
Krsmanovic
Korisničko ime
dusan1995
Email
duci1995@gmail.com
Aktivnost
1
Status
Nije obrisan
Datum registracije
2019-01-Thu 11:57:47

Ime
Dušan
Prezime
Krsmanović
Korisničko ime
dusan195
Email
duci1995@gmail.com
Aktivnost
1
Status
Nije obrisan
Datum registracije
25.02.2019

Ime
Dušan
Prezime
Krsmanović
Korisničko ime
dusan19950
Email
duci1995@gmail.com
Aktivnost
1
Status
Nije obrisan
Datum izmene
07.03.2019 18:22:31

Ime
Dušan
Prezime
Krsmanović
Korisničko ime
dusan19995
Email
duci1995@gmail.com
Aktivnost
1
Status
Nije obrisan
Datum izmene
07.03.2019 20:04:29

Ime
Dušan
Prezime
Krsmanović
Korisničko ime
gzzuututz
Email
duci1995@gmail.com
Aktivnost
0
Status
Nije obrisan
Datum registracije
14.03.2019

Ime
Dušan
Prezime
Krsmanović
Korisničko ime
fdgdfgdfgd
Email
duci1995@gmail.com
Aktivnost
0
Status
Nije obrisan
Datum registracije
14.03.2019

Ime
Dušan
Prezime
Krsmanović
Korisničko ime
gddfgdgd
Email
duci1995@gmail.com
Aktivnost
0
Status
Nije obrisan
Datum registracije
14.03.2019

Ime
Dušan
Prezime
Krsmanović
Korisničko ime
du345353
Email
duci1995@gmail.com
Aktivnost
0
Status
Nije obrisan
Datum registracije
14.03.2019

Ime
Dušan
Prezime
Krsmanović
Korisničko ime
uhizi7667
Email
duci1995@gmail.com
Aktivnost
0
Status
Nije obrisan
Datum registracije
14.03.2019

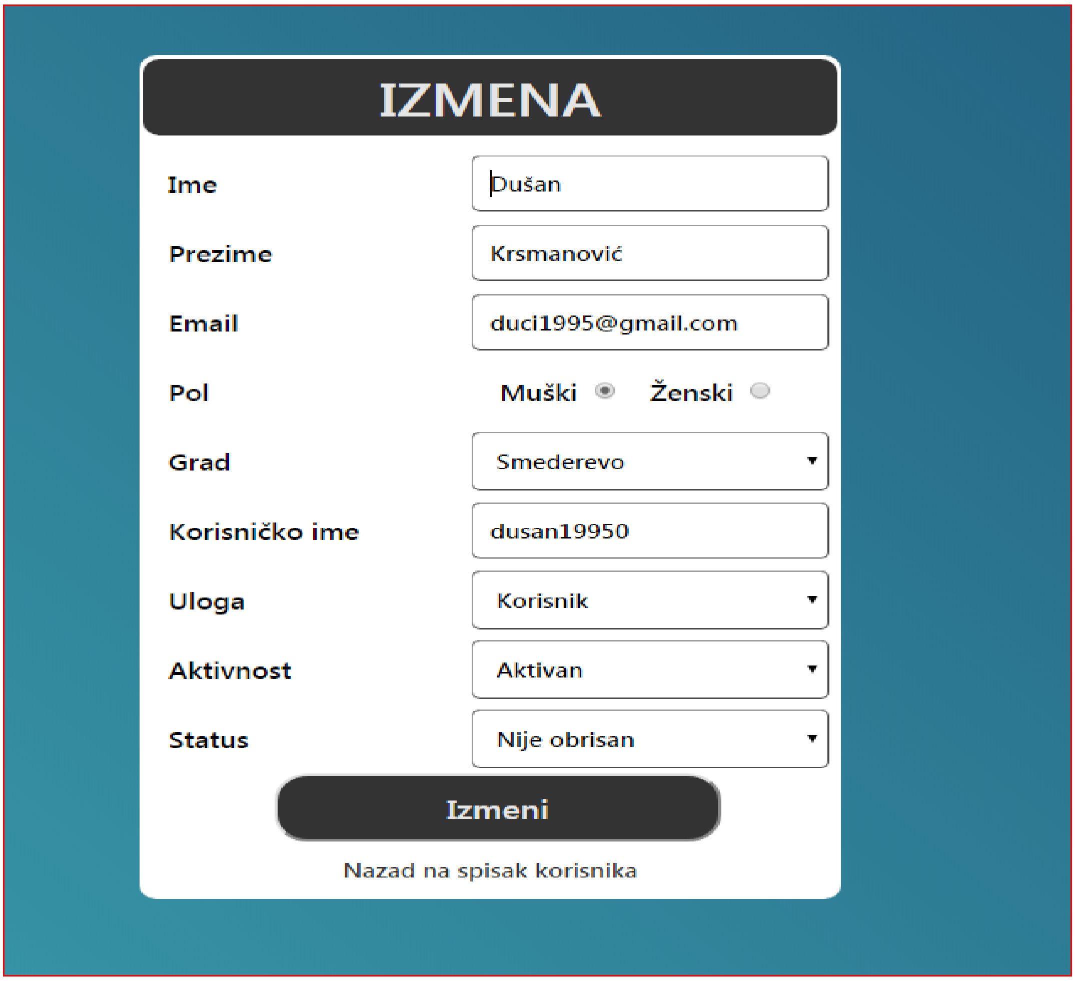
Ime
Dušan
Prezime
Krsmanović
Korisničko ime
zuzuzu76
Email
duci1995@gmail.com
Aktivnost
0
Status
Nije obrisan
Datum registracije
14.03.2019

Ime
Dušan
Prezime
Krsmanović
Korisničko ime
du47543dfu
Email
duci1995@gmail.com
Aktivnost
0
Status
Nije obrisan
Datum registracije
14.03.2019

Ime
Dušan
Prezime
Krsmanović
Korisničko ime
fdsjfojo34
Email
duci1995@gmail.com
Aktivnost
0
Status
Nije obrisan
Datum registracije
14.03.2019

Ime
Dušan
Prezime
Krsmanović
Korisničko ime
jiturnfe3
Email
duci1995@gmail.com
Aktivnost
1
Status
Nije obrisan
Datum registracije
14.03.2019

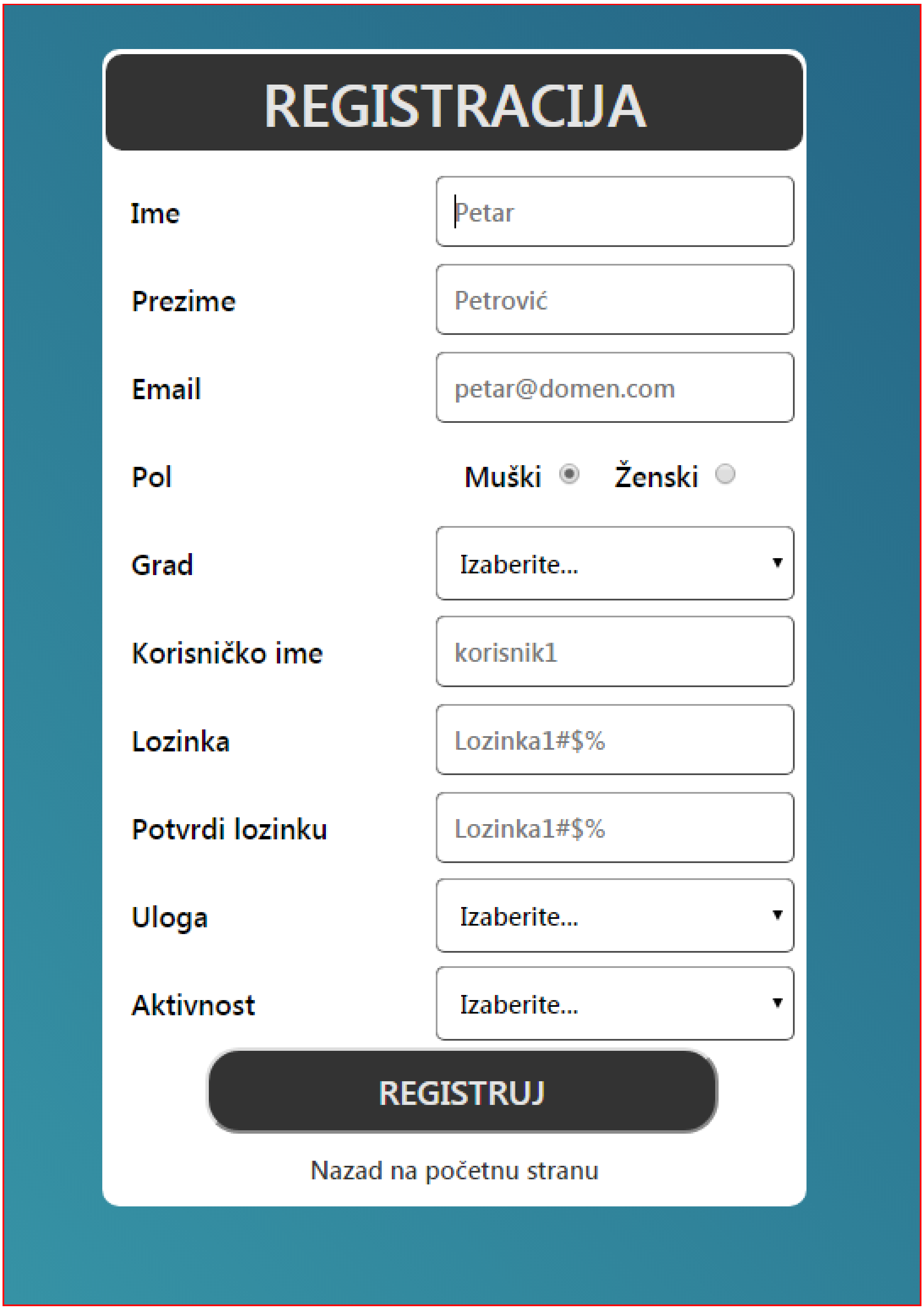
admin/korisnci/forma-izmena



IZMENA

Ime	Dušan
Prezime	Krsmanović
Email	duci1995@gmail.com
Pol	Muški ● Ženski ○
Grad	Smederevo
Korisničko ime	dusan19950
Uloga	Korisnik
Aktivnost	Aktivan
Status	Nije obrisan

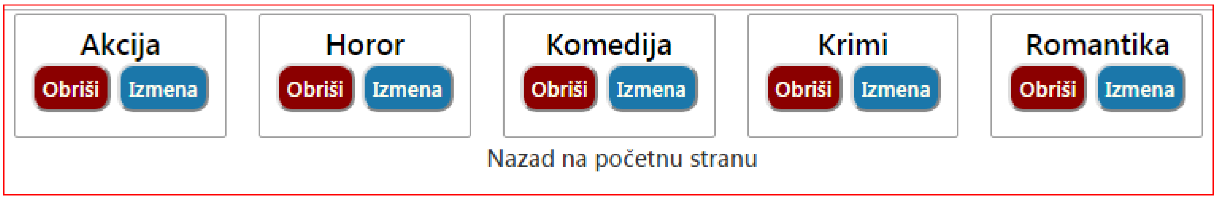
admin/korisnici/forma



REGISTRACIJA

Ime	Petar
Prezime	Petrović
Email	petar@domen.com
Pol	Muški ● Ženski ○
Grad	Izaberite... ▾
Korisničko ime	korisnik1
Lozinka	Lozinka1#$%
Potvrdi lozinku	Lozinka1#$%
Uloga	Izaberite... ▾
Aktivnost	Izaberite... ▾

admin/zanrovi/spisak



Akcija

Horor

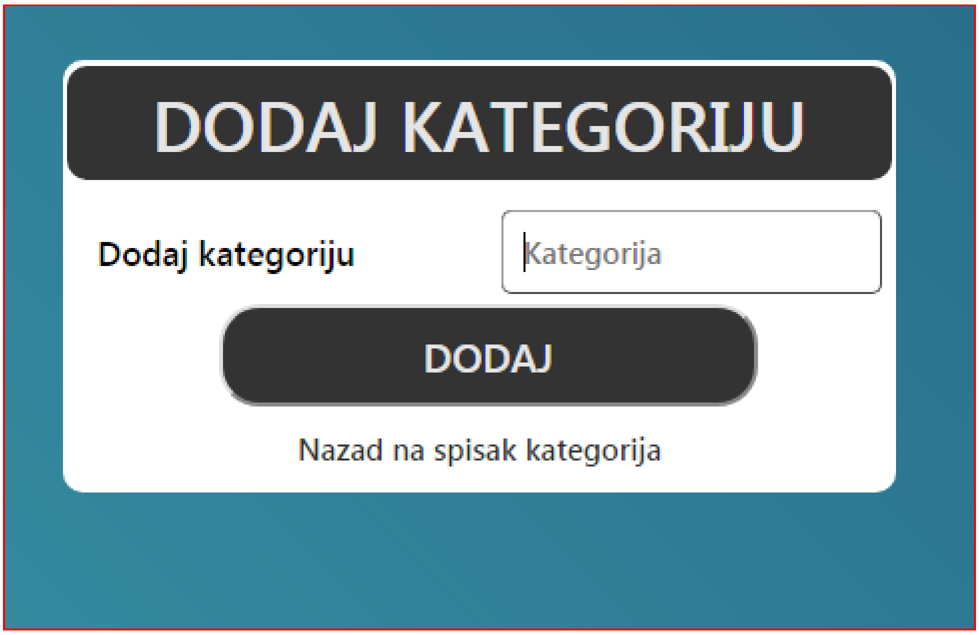
Komedija

Krimi

Romantika

Nazad na početnu stranu

admin/zanrovi/forma

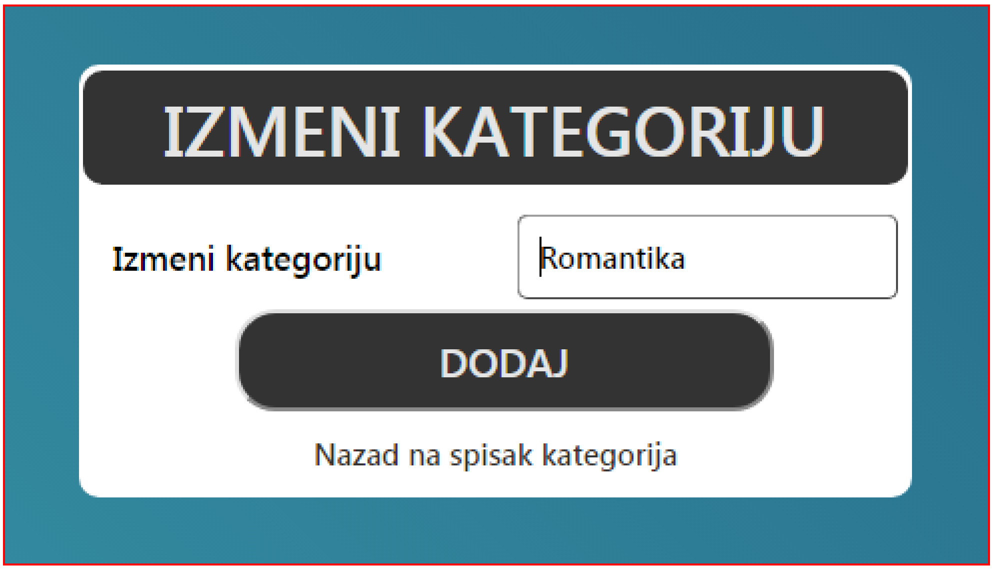


DODAJ KATEGORIJU

Dodaj kategoriju: Kategorija

Nazad na spisak kategorija

admin/zanrovi/forma-izmena

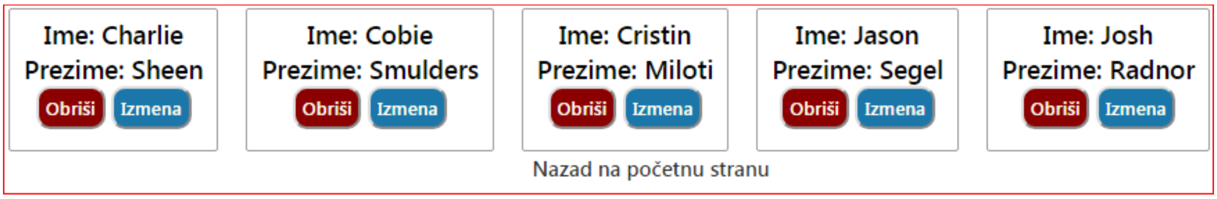


IZMENI KATEGORIJU

Izmeni kategoriju: Romantika

Nazad na spisak kategorija

admin/uloge/spisak



Ime: Charlie
Prezime: Sheen

Ime: Cobie
Prezime: Smulders

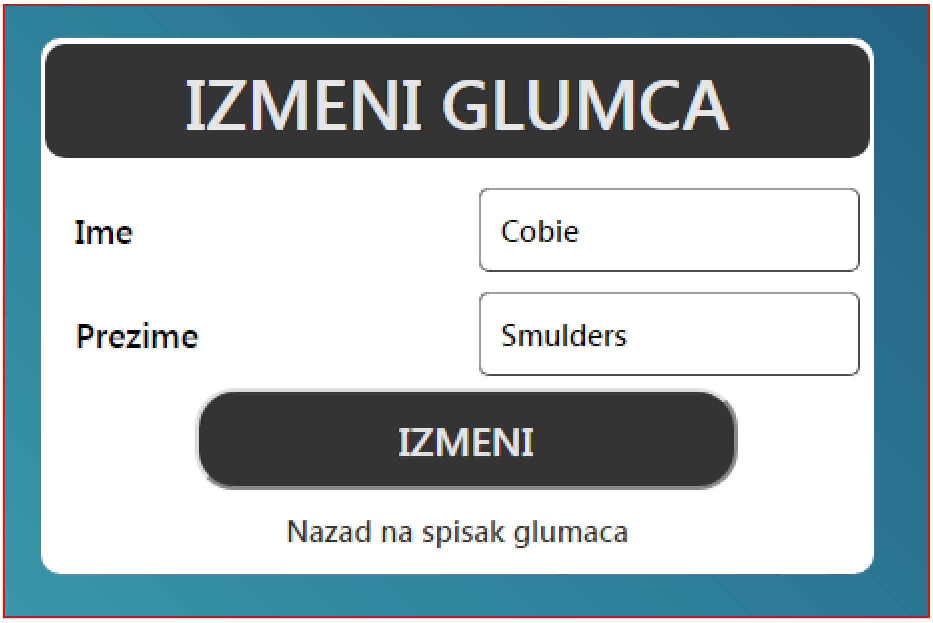
Ime: Cristin
Prezime: Miloti

Ime: Jason
Prezime: Segel

Ime: Josh
Prezime: Radnor

Nazad na početnu stranu

admin/uloge/forma-izmena



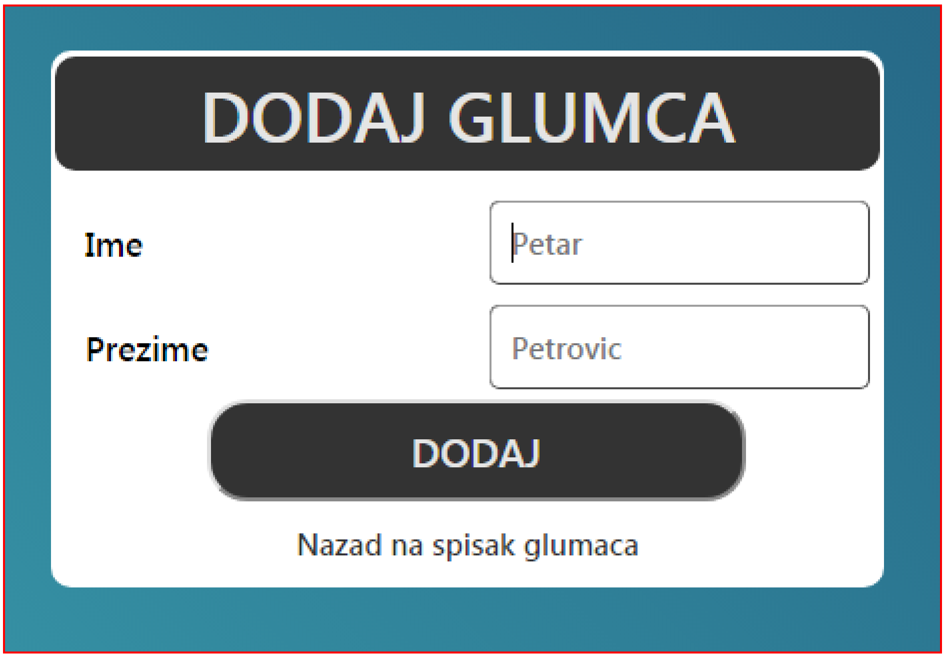
IZMENI GLUMCA

Ime: Cobie

Prezime: Smulders

Nazad na spisak glumaca

admin/uloge/forma



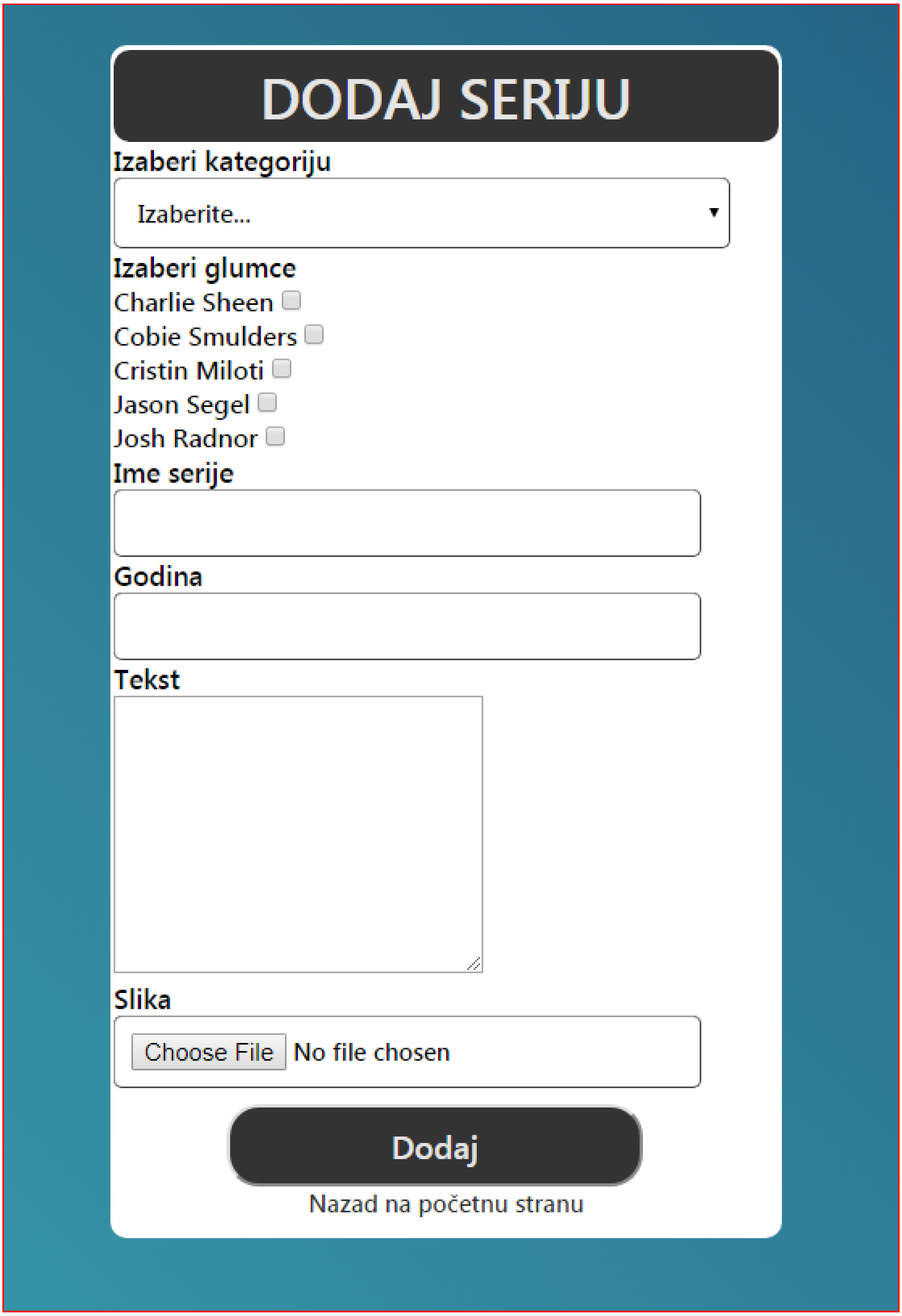
DODAJ GLUMCA

Ime: Petar

Prezime: Petrovic

Nazad na spisak glumaca

admin/serije/forma



DODAJ SERIJU

Izaberi kategoriju

Izaberite... ▾

Izaberi glumce

Charlie Sheen □

Cobie Smulders □

Cristin Miloti □

Jason Segel □

Josh Radnor □

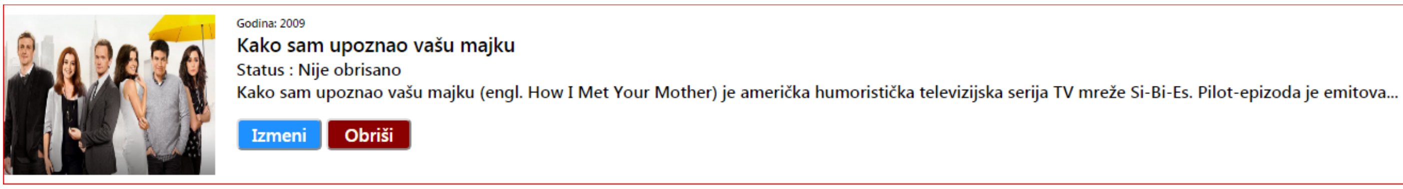
Ime serije

Godina

Tekst

Slika

admin/serije/spisak



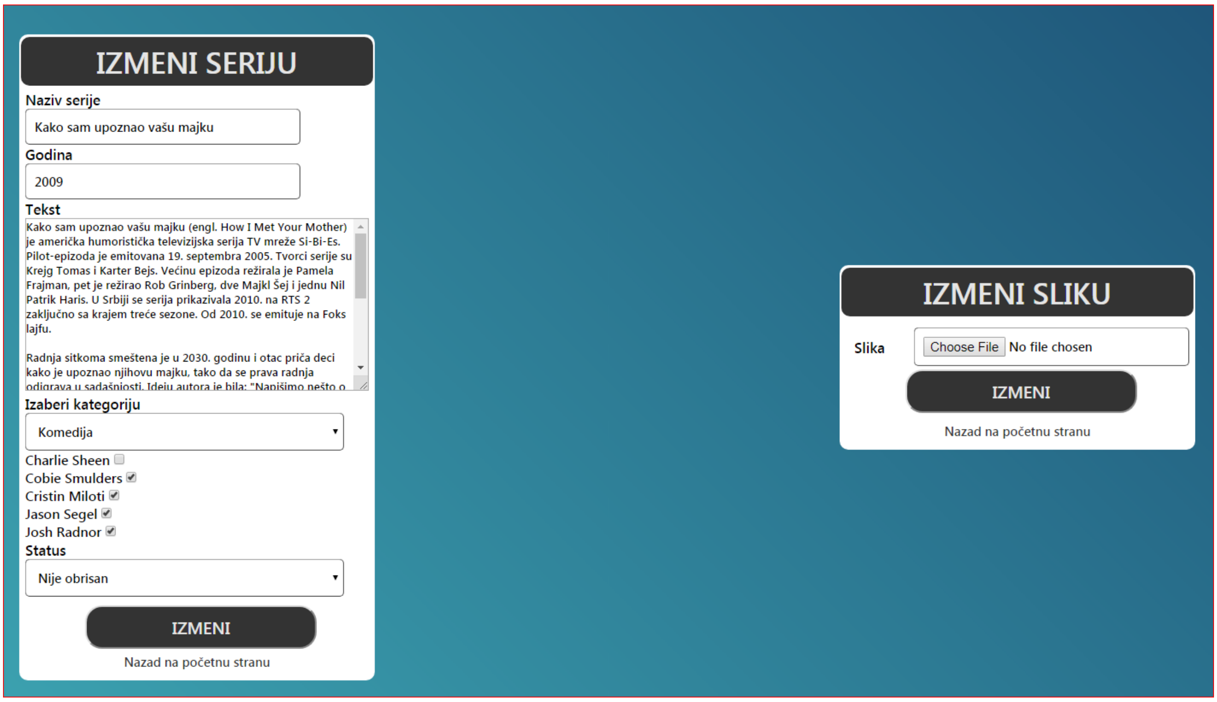
Godina: 2009

Kako sam upoznao vašu majku

Status : Nije obrisano

Kako sam upoznao vašu majku (engl. How I Met Your Mother) je američka humoristička televizijska serija TV mreže Si-Bi-Es. Pilot-epizoda je emitova...

admin/serije/forma-izmena



IZMENI SERIJU

Naziv serije: Kako sam upoznao vašu majku

Godina: 2009

Tekst:

Kako sam upoznao vašu majku (engl. How I Met Your Mother) je američka humoristička televizijska serija TV mreže Si-Bi-Es. Pilot-epizoda je emitovana 19. septembra 2005. Tvorci serije su Krejg Tomas i Karter Bejs. Većinu epizoda režirala je Pamela Frajman, pet je režirao Rob Grinberg, dve Majkl Šej i jednu Nil Patrik Haris. U Srbiji se serija prikazivala 2010. na RTS 2 zaključno sa krajem treće sezone. Od 2010. se emituje na Foks lajfu.

Radnja sitkoma smeštena je u 2030. godinu i otac priča deci kako je upoznao njihovu majku, tako da se prava radnja odigrava u sadašnjosti. Ideju autora je bila: "Napišimo nešto o

Izaberi kategoriju: Komedija

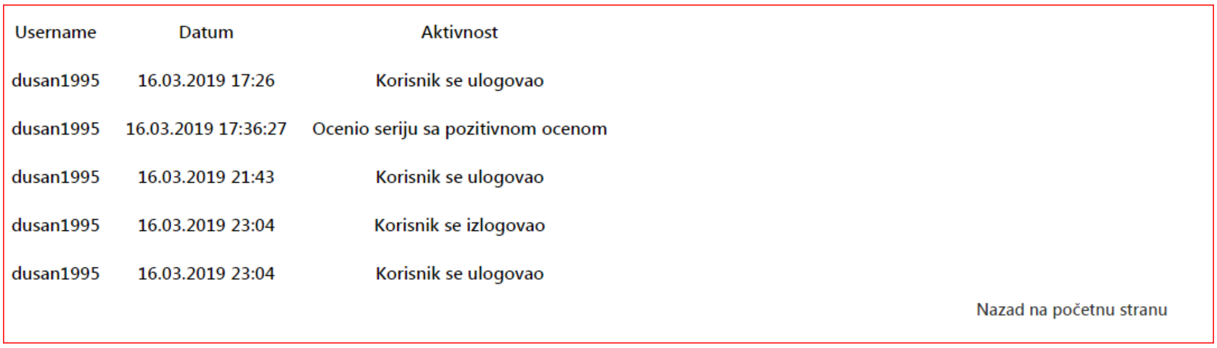
Charlie Sheen ☐
Cobie Smulders ☑
Cristin Miloti ☑
Jason Segel ☑
Josh Radnor ☑

Status: Nije obrisan

IZMENI SLIKU

Slika: Choose File No file chosen

admin/izvestaj



Username	Datum	Aktivnost
dusan1995	16.03.2019 17:26	Korisnik se ulogovao
dusan1995	16.03.2019 17:36:27	Ocenio seriju sa pozitivnom ocenom
dusan1995	16.03.2019 21:43	Korisnik se ulogovao
dusan1995	16.03.2019 23:04	Korisnik se izlogovao
dusan1995	16.03.2019 23:04	Korisnik se ulogovao

Kontroleri

UserFeedbackController
feedback(FeedBackRequest $reqeust)

SeriesController
__construct()
newest()
dislike(Request $request)
like(Request $request)
show($id)
search($char)

LogRegController
__construct()
login(LogRequest $request)
register(RegRequest $request)
activation($param)
logout(Request $request)

FrontendController
__construct()
login()
register()
kontakt()
autor()

CommentsController
__construct()
store(SendCommentRequest $request)
showSeriesComments($idseries)
delete(Request $request)
like(Request $request)
dislike(Request $request)

CategoriesController
__construct()
show($cat)

Admin/ActivityController
index()

Admin/GenreController
__construct()
index()
create()
store(GenreRequest $request)
edit($id)
update(GenreRequest $request, $id)
destroy($id)

Admin/RoleController
__construct()
index()
create()
store(Request $request)
edit($id)
update(Request $request, $id)
destroy($id)

Admin/SeriesController
__construct()
index()
store(AdminInsertSeriesRequest $request)
edit($id, $cat)
destroy($id)
updateInfo($id, Request $request)
updatePicture($id, PictureRequest $picture)

Admin/SponsorController
__construct()
index()
create()
store(SponsorRequest $request, PictureRequest $picture)
edit($id)
updateLink(SponsorRequest $request, $id)
updatePicture(PictureRequest $pictureRequest, $id)

destroy($id)

Admin/UserController
__construct()
index()
create()
store(AdminInsertUserRequest $request)
edit($id)
update(Request $request, $id)
destroy($id)

Modeli

Glumci
getAll()
delete()
update()
insert()
getOne()

Grad
getAll()

Izveštaj
insert()
getAll()

Komentar
getSeriesComments()
insert()
delete()
like()
dislike()

Korisnik
login()
register()
getUsers()
insert()
getOneUser()
update()
delete()
activate()

Link
getMenu()
getCategories()
insert()
delete()
getOne()
update()
getOtherCategories()

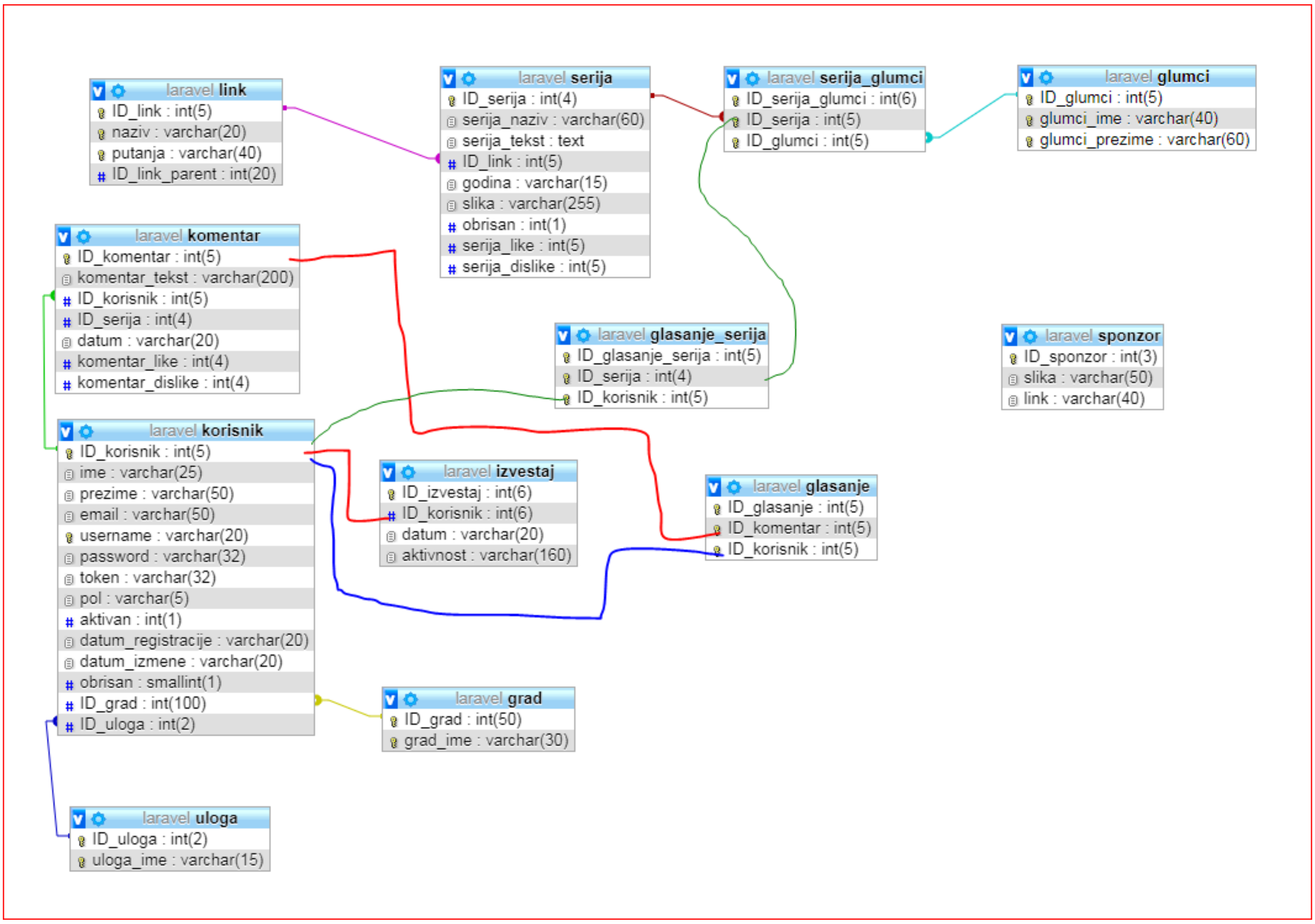
Serija
getAll()
getSeriesByCategory()
getNewestSeries()
dislike()
like()
getOne()
getRoles()

search()
delete()
insert()
insertCredits()
updateInfo()
updateCreditsFirstStep()
updateCreditsFinalStep()
updatePicture()

Sponzor
getSponsors()
insertSponsor()
deleteSponsor()
getOneSponsor()
updateLink()
updatePicture()

Uloga
getRoles()

Dizajn baze



Kodovi

------------------------------routes/web.php---------------------------

```
<?php

/*
|--------------------------------------------------------------------------
| Web Routes
|--------------------------------------------------------------------------
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|
*/
Route::pattern('id','[0-9]+');
```

```
Route::pattern('cat','[\w]+');
Route::pattern('char','[\w]+');

Route::get('/prijava',"FrontendController@login")->name("renderLogin");
Route::get('/registracija',"FrontendController@register")->name("renderRegistration");

Route::get("/logout", "LogRegController@logout")->name("logout");

Route::post('/login','LogRegController@login');
Route::post('/register','LogRegController@register');

Route::get('/activation/{param}','LogRegController@activation')->name('activation');

Route::get ("/kontakt","FrontendController@kontakt")->name("kontakt");
Route::get ("/autor","FrontendController@autor")->name("autor");

Route::post("/feedback","UserFeedbackController@feedback");
Route::get ("/","SeriesController@newest")->name("pocetna");
Route::get ("/serija/{id}","SeriesController@show");
Route::get("/{cat}","CategoriesController@show")->name('category');
Route::get("/pretraga/{char}","SeriesController@search");

Route::get("/komentari/{idseries}","CommentsController@showSeriesComments");


Route::group(['middleware' => 'session'] , function() {
    Route::post('/komentari', 'CommentsController@store');

    Route::post("/seriesDislike", "SeriesController@dislike");
    Route::post("/seriesLike", "SeriesController@like");

    Route::post('/commentsLike', "CommentsController@like");
    Route::post('/commentsDislike', "CommentsController@dislike");
    Route::delete('/commentsDelete', 'CommentsController@delete');

});

Route::group(['middleware' => 'admin'] , function(){

    Route::get("/admin/sponzori/spisak","Admin\SponsorController@index")->name("sponsor-show");
    Route::get("/admin/sponzori/forma","Admin\SponsorController@create")->name("sponsor-form");
    Route::get("/admin/sponzori/forma-izmena/{id}","Admin\SponsorController@edit")->name("sponsor-one-render");
    Route::post("/admin/sponzori/unos","Admin\SponsorController@store")->name("sponsor-insert");

Route::delete("/admin/sponzori/brisanje/{id}","Admin\SponsorController@destroy");

Route::put("/admin/sponzori/izmena/link/{id}","Admin\SponsorController@updateLink")->name("sponsor-update-link");

Route::put("/admin/sponzori/izmena/slika/{id}","Admin\SponsorController@updatePicture")->name("sponsor-update-picture");

    Route::get("/admin/korisnici/spisak","Admin\UserController@index")->name("user-show");
    Route::get("/admin/korisnici/forma","Admin\UserController@create")->name("user-form");
    Route::post("/admin/korisnici/unos","Admin\UserController@store");
    Route::get("/admin/korisnici/forma-izmena/{id}","Admin\UserController@edit")->name("user-one-render");
    Route::post("/admin/korisnici/izmena/{id}","Admin\UserController@update");

Route::delete("/admin/korisnici/brisanje/{id}","Admin\UserController@destroy");

    Route::get("/admin/zanrovi/spisak","Admin\GenreController@index")->name("genre-show");
    Route::get("/admin/zanrovi/forma","Admin\GenreController@create")->name("genre-form");
    Route::get("/admin/zanrovi/forma/{id}","Admin\GenreController@edit")->name("genre-one-render");
    Route::post("/admin/zanrovi/izmena/{id}","Admin\GenreController@update");

Route::delete("/admin/zanrovi/brisanje/{id}","Admin\GenreController@destroy");
    Route::post("/admin/zanrovi/unos","Admin\GenreController@store");

    Route::get("/admin/uloge/spisak","Admin\RoleController@index")->name('role-show');
    Route::delete("/admin/uloge/brisanje/{id}","Admin\RoleController@destroy");
    Route::get("/admin/uloge/forma","Admin\RoleController@create")->name("role-form");
    Route::post("/admin/uloge/unos","Admin\RoleController@store")->name("role-insert");
    Route::get("/admin/uloge/forma-izmena/{id}","Admin\RoleController@edit")->name("role-one-render");
    Route::put("/admin/uloge/izmena/{id}","Admin\RoleController@update");
    Route::get("/admin/serije/spisak",'Admin\SeriesController@index')->name("series-show");
    Route::get("/admin/serije/forma",'Admin\SeriesController@create')->name('series-form');
    Route::get("/admin/serije/forma-izmena/{id}/{cat}","Admin\SeriesController@edit")->name("series-one-render");
    Route::delete('/admin/serije/brisanje/{id}','Admin\SeriesController@destroy');
    Route::put('/admin/serije/izmena-info/{id}','Admin\SeriesController@updateInfo')->name('series-update-info');
    Route::put('/admin/serije/izmena-slike/{id}','Admin\SeriesController@updatePicture')->name('series-update-picture');
    Route::post('/admin/serije/unos','Admin\SeriesController@store')->name('series-insert');

    Route::get('/admin/izvestaj','Admin\ActivityController@index')->name('activity');


});
```

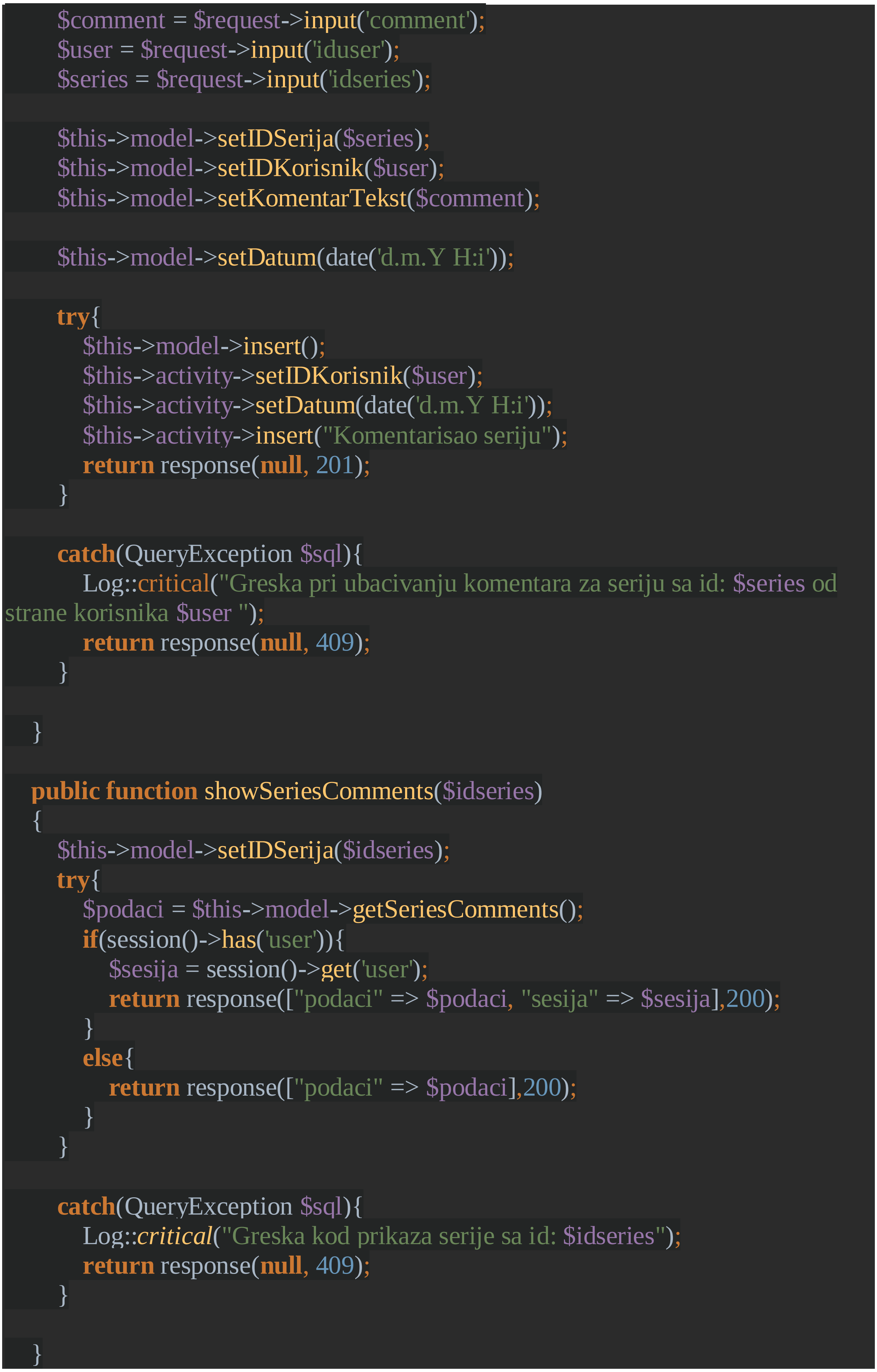
---------------------------------Controllers--

CategoriesController.php

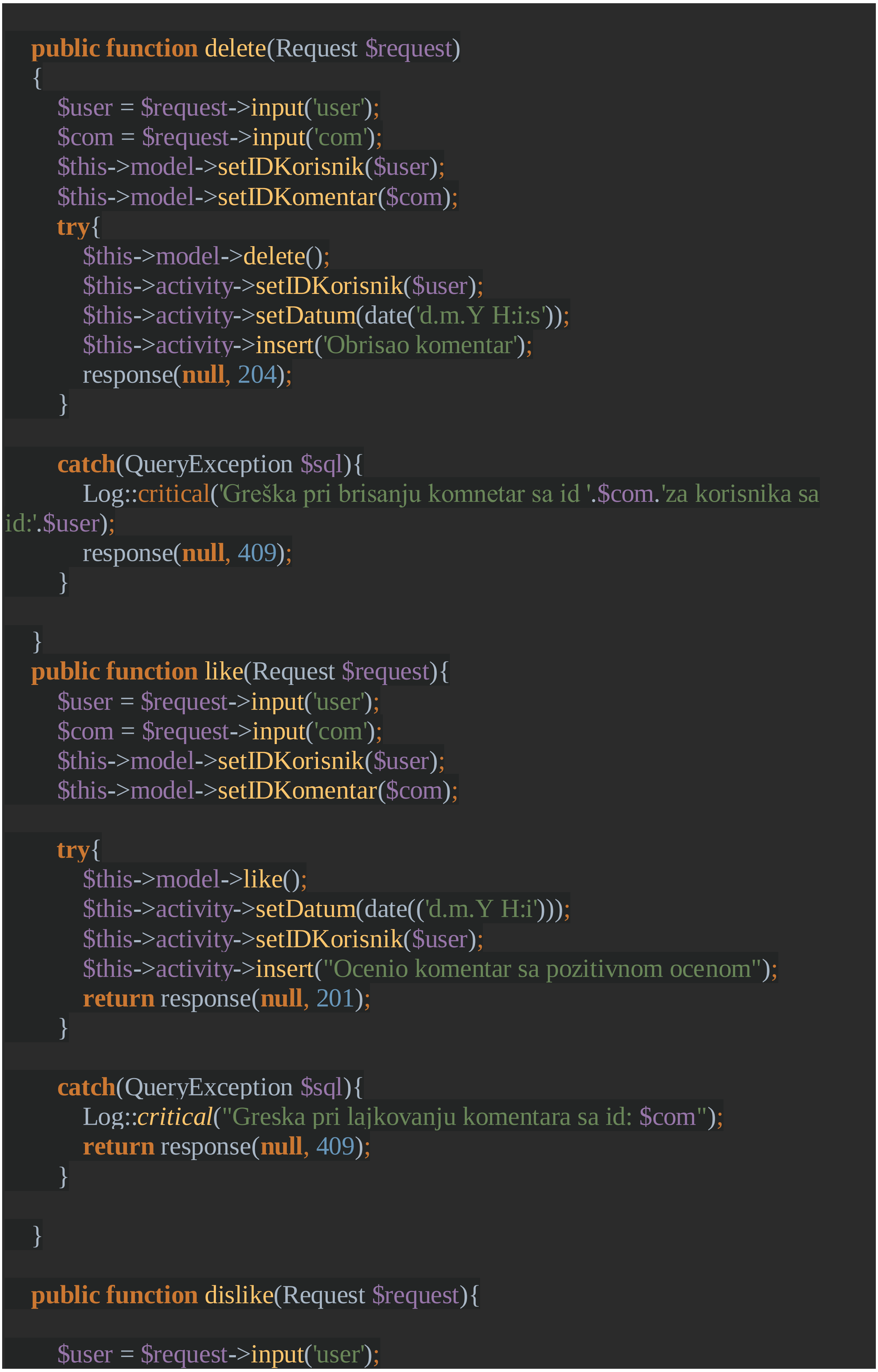
```
<?php

namespace App\Http\Controllers;

use App\Models\Link;
use App\Models\Serija;
use Illuminate\Database\QueryException;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;

class CategoriesController extends FrontendController
{
    public function __construct()
    {
        parent::__construct();
        $this->model = new Serija();
    }

    public function show($cat)
    {

        $this->model->setPutanja($cat);
        $category = $this->model->getSeriesByCategory();
        return view("pages.serije",$this->data)->with("serije", $category);
    }



}
```

CommentsController.php

```
<?php

namespace App\Http\Controllers;

use App\Http\Requests\SendCommentRequest;
use App\Models\Izvestaj;
use App\Models\Komentar;
use Illuminate\Database\QueryException;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;

class CommentsController extends Controller
{
    private $model;
    private $activity;

    public function __construct()
    {
        $this->activity = new Izvestaj();
        $this->model = new Komentar();
    }



    /**
     * Store a newly created resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @return \Illuminate\Http\Response
     */
    public function store(SendCommentRequest $request)
    {
```



```
        $comment = $request->input('comment');
        $user = $request->input('iduser');
        $series = $request->input('idseries');

        $this->model->setIDSerija($series);
        $this->model->setIDKorisnik($user);
        $this->model->setKomentarTekst($comment);

        $this->model->setDatum(date('d.m.Y H:i'));

        try{
            $this->model->insert();
            $this->activity->setIDKorisnik($user);
            $this->activity->setDatum(date('d.m.Y H:i'));
            $this->activity->insert("Komentarisao seriju");
            return response(null, 201);
        }

        catch(QueryException $sql){
            Log::critical("Greska pri ubacivanju komentara za seriju sa id: $series od
strane korisnika $user ");
            return response(null, 409);
        }

    }

    public function showSeriesComments($idseries)
    {
        $this->model->setIDSerija($idseries);
        try{
            $podaci = $this->model->getSeriesComments();
            if(session()->has('user')){
                $sesija = session()->get('user');
                return response(["podaci" => $podaci, "sesija" => $sesija],200);
            }
            else{
                return response(["podaci" => $podaci],200);
            }
        }

        catch(QueryException $sql){
            Log::critical("Greska kod prikaza serije sa id: $idseries");
            return response(null, 409);
        }

    }
```



```
    public function delete(Request $request)
    {
        $user = $request->input('user');
        $com = $request->input('com');
        $this->model->setIDKorisnik($user);
        $this->model->setIDKomentar($com);
        try{
            $this->model->delete();
            $this->activity->setIDKorisnik($user);
            $this->activity->setDatum(date('d.m.Y H:i:s'));
            $this->activity->insert('Obrisao komentar');
            response(null, 204);
        }

        catch(QueryException $sql){
            Log::critical('Greška pri brisanju komnetar sa id '.$com.'za korisnika sa
id:'.$user);
            response(null, 409);
        }

    }
    public function like(Request $request){
        $user = $request->input('user');
        $com = $request->input('com');
        $this->model->setIDKorisnik($user);
        $this->model->setIDKomentar($com);

        try{
            $this->model->like();
            $this->activity->setDatum(date(('d.m.Y H:i')));
            $this->activity->setIDKorisnik($user);
            $this->activity->insert("Ocenio komentar sa pozitivnom ocenom");
            return response(null, 201);
        }

        catch(QueryException $sql){
            Log::critical("Greska pri lajkovanju komentara sa id: $com");
            return response(null, 409);
        }

    }

    public function dislike(Request $request){

        $user = $request->input('user');
        $com = $request->input('com');
        $this->model->setIDKorisnik($user);
        $this->model->setIDKomentar($com);

        try{
            $this->model->dislike();
            $this->activity->setDatum(date(('d.m.Y H:i')));
            $this->activity->setIDKorisnik($user);
            $this->activity->insert("Ocenio komentar sa negativnom ocenom");

            return response(null, 201);
        }

        catch(QueryException $sql){
            Log::critical("Greska pri dislajkovanju komentara sa id: $com");
            return response(null, 409);
        }

    }
}
```

FrontendController.php

```
<?php
namespace App\Http\Controllers;
use App\Models\Grad;
use App\Models\Link;
use App\Models\Serija;
use App\Models\Sponzor;
class FrontendController extends Controller
{
    protected $data;
    protected $model;
    public function __construct(){
        $link = new Link();
        $this->data['kategorije'] = $link->getCategories();
        $this->data['meni'] = $link->getMenu();
        $sponzori = new Sponzor();
        $this->data['sponzori'] = $sponzori->getSponsors();
        $this->data['linkovi'] = $link->getMenu();
    }
    public function login(){
        return view('pages.prijava');
    }
    public function register(){
        $city = new Grad();
        $data = $city->getAll();
        return view('pages.registracija')->with("cities", $data);
    }

    public function kontakt(){
        return view("pages.kontakt",$this->data);
    }
    public function autor(){
        return view('pages.autor',$this->data);
    }
}
```

LogRegController.php

```
<?php
namespace App\Http\Controllers;
use App\Http\Requests\LogRequest;
use App\Http\Requests\RegRequest;
use App\Models\Izvestaj;
use App\Models\Korisnik;
use Illuminate\Database\QueryException;
use Illuminate\Http\Request;
use App\Http\Requests\LoginRequest;
use Illuminate\Support\Facades\Log;
use PHPMailer\PHPMailer\Exception;
use PHPMailer\PHPMailer\PHPMailer;
use Psy\Util\Json;
class LogRegController extends Controller
{    /**
     * LogRegController constructor.
     */
    private $user;
    private $date;
    public function __construct()
    {
        $this->user = new Korisnik();
        $this->date = date('d.m.Y H:i');
    }
    public function login(LogRequest $request){
        $this->user->setUsername($request->input("username"));
        $this->user->setPassword($request->input("password"));

        try {
            $us = $this->user->login();
            if($us != null) {
                $request->session()->put("user", $us);
                $model = new Izvestaj();
                $model->setDatum($this->date);
                $model->setIDKorisnik(session()->get('user')->ID_korisnik);
                $model->insert("Korisnik se ulogovao");
                return  response(null,200);
            }
            else
                return response(null,401);
        }
        catch(QueryException $sql){
            Log::critical("Greska u radu sa bazom pri logovanju korisnika");
            return response(null, 500);
        }
    }
    public function register(RegRequest $request) {

        $this->user->setIme($request->input('firstname'));
        $this->user->setPrezime($request->input('lastname'));
        $this->user->setDatumRegistracije(date("d.m.Y"));
        $token = md5(time());
        $this->user->setToken($token);
        $this->user->setUsername($request->input("username"));
        $this->user->setPassword($request->input("password"));
        $this->user->setPol($request->input("gender"));
        $this->user->setIDGrad($request->input("city"));
        $email = $request->input("email");
        $this->user->setEmail($email);


            $mail = new PHPMailer(true);
        try {
            $this->user->register();
            //Server settings
            //$mail->SMTPDebug = 2;                                 // Enable verbose debug output
            $mail->isSMTP();                                      // Set mailer to use SMTP
            $mail->Host = 'smtp.gmail.com';  // Specify main and backup SMTP servers

            $mail->SMTPAuth = true;
            $mail->SMTPSecure = true;
            // Enable SMTP authentication
            $mail->Username = 'phpmailer1995@gmail.com';                 // SMTP username
            $mail->Password = 'dusan1995';                           // SMTP password
            $mail->SMTPSecure = 'tls';                              // Enable TLS encryption, `ssl` also accepted
            $mail->Port = 587;                                      // TCP port to connect to

            $mail->SMTPOptions = array(
                'ssl' => array(
                    'verify_peer' => false,
                    'verify_peer_name' => false,
                    'allow_self_signed' => true
                )
            );
            //Recipients
            $mail->setFrom('phpmailer1995@gmail.com', 'Registracija');
            $mail->addAddress($email);     // Add a recipient

            //Content
            $mail->isHTML(true);                                  // Set email format to HTML
            $mail->Subject = 'Aktivirajte nalog';

            $mail->Body = 'Klinkite na <a href="http://127.0.0.1:8000/activation/' . $token . '" >link</a> da aktivirate profil';

            $mail->send();

            return response(null, 201);
        }

        catch(QueryException $exception){
            Log::error($exception->getMessage());
            return response(null,409);
        }
        catch (Exception $e) {
            Log::critical('Greska pri slanju mejla za registraciju');
            return response(null, 500);
        }
    }

    public function activation($param) {

        $this->user->setToken($param);
        try{
            $this->user->activate();
            return redirect('/prijava');
        }
        catch(QueryException $sql){
            Log::critical("Greska pri aktivaciji korisnika sa tokenom $param");
            return redirect()->back();
        }
    }

    public function logout(Request $request){
        $model = new Izvestaj();
        $model->setIDKorisnik(session()->get('user')->ID_korisnik);
        $model->setDatum($this->date);
        try{
            $model->insert("Korisnik se izlogovao");
        }
        catch (QueryException $sql){
            Log::error('Greska pri ubacivanju u tabelu izvestaj');
        }
        $request->session()->forget("user");
        $request->flush();
        return redirect(route("renderLogin"));
    }

}
```

SeriesController.php

```
<?php

namespace App\Http\Controllers;

use App\Models\GlasanjeSerija;
use App\Models\Izvestaj;
use App\Models\Serija;
use Illuminate\Database\QueryException;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;
use Psy\Util\Json;


class SeriesController extends FrontendController
{
    private $date;
    private $activity;
    public function __construct()
    {
        parent::__construct();
        $this->model = new Serija();
        $this->date = date("d.m.Y H:i:s");
        $this->activity = new Izvestaj();
    }

    public function newest()
    {
        $serija = $this->model->getNewestSeries();
        return view('pages.pocetna', $this->data)->with("serije", $serija);
    }

    public function dislike(Request $request)
    {
        $user = $request->input("user");
        $series = $request->input("series");
        $this->model->setIDSerija($series);
        $this->model->setIDKorisnik($user);
        try {
            $this->model->dislike();
            $this->activity->setIDKorisnik($user);
            $this->activity->setDatum($this->date);
            $this->activity->insert("Ocenio seriju sa negativnom ocenom");
            return response(null, 204);
        } catch (QueryException $sql) {
            Log::critical("Greska pri dislajku serije " . date("d-m-Y H:i:s") . " ID_serije
" . $series);
            return response(null, 409);
        }
    }

    public function like(Request $request)
    {
        $user = $request->input("user");
        $series = $request->input("series");
        $this->model->setIDSerija($series);
        $this->model->setIDKorisnik($user);

        try {
            $this->model->like();
            $this->activity->setIDKorisnik($user);
            $this->activity->setDatum($this->date);
            $this->activity->insert("Ocenio seriju sa pozitivnom ocenom");
            return response(null, 204);
        } catch (QueryException $sql) {
            Log::critical("Greska pri lajku serije " . date("d-m-Y H:i:s") . " ID_serije " .
$series);
            return response(null, 409);
        }
    }

    public function show($id){
        $this->model->setIDSerija($id);
        try {
            $podaci = $this->model->getOne();
            $glumci = $this->model->getRoles();
            if (session()->has('user')) {
                $user = session()->get("user");
                return response(["podaci" => $podaci, "glumci" => $glumci ,"korisnik" => $user], 200);
            }
            else
                return response(["podaci" => $podaci, "glumci" => $glumci], 200);
        }
        catch (QueryException $sql) {
            Log::critical("Greska pri dohvatnaju infomracija za seriju sa id: $id na dan $this->date");
            return response(null, 409);
        }
    }
    public function search($char){

        $this->model->setSerijaNaziv($char);
        try{
            $podaci = $this->model->search();
            if (session()->has('user')) {
                $user = session()->get("user");
                return response(["podaci" => $podaci, "korisnik" => $user], 200);
            }
            else
                return response(["podaci" => $podaci], 200);
        }
        catch(QueryException $sql){
            Log::critical("Greska pri pretrazi sa kriterijumom $char na dan $this->date");
            return response(null,404);
        }
    }
}
```

UserFeedbackController.php

```
<?php

```

```
namespace App\Http\Controllers;

use App\Http\Requests\FeedbackRequest;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;
use PHPMailer\PHPMailer\PHPMailer;

class UserFeedbackController extends Controller
{
    public function feedback(FeedbackRequest $request){

        $first = $request->input("ime");
        $last = $request->input("prezime");
        $email = $request->input("email");
        $content = $request->input("content");

        $mail = new PHPMailer(true);
        try {

            //Server settings
//          $mail->SMTPDebug = 2;                                 // Enable verbose debug
output
            $mail->isSMTP();                                      // Set mailer to use SMTP
            $mail->Host = 'smtp.gmail.com';  // Specify main and backup SMTP servers

            $mail->SMTPAuth = true;
            $mail->SMTPSecure = true;
            // Enable SMTP authentication
            $mail->Username = 'phpmailer1995@gmail.com';                 // SMTP
username
            $mail->Password = 'dusan1995';                           // SMTP password
            $mail->SMTPSecure = 'tls';                            // Enable TLS encryption, `ssl`
also accepted
            $mail->Port = 587;                                    // TCP port to connect to

            $mail->SMTPOptions = array(
                'ssl' => array(
                    'verify_peer' => false,
                    'verify_peer_name' => false,
                    'allow_self_signed' => true
                )
            );
            //Recipients
            $mail->setFrom('phpmailer1995@gmail.com', 'Pitanje');
            $mail->addAddress("duci1995@gmail.com");     // Add a recipient
            //Content
            $mail->isHTML(true);                                  // Set email format to HTML
            $mail->Subject = 'Pitanje';

            $mail->Body= "$first $last <br/> $email<br/>$content";

            $mail->send();

            return response(null,200);

        } catch (\Exception $e) {
            Log::debug("Greška pri slanju mejla za kontakt");
            return response(null ,500);
        }

    }
}
```

--------------------------------Controllers/Admin--------------------------------

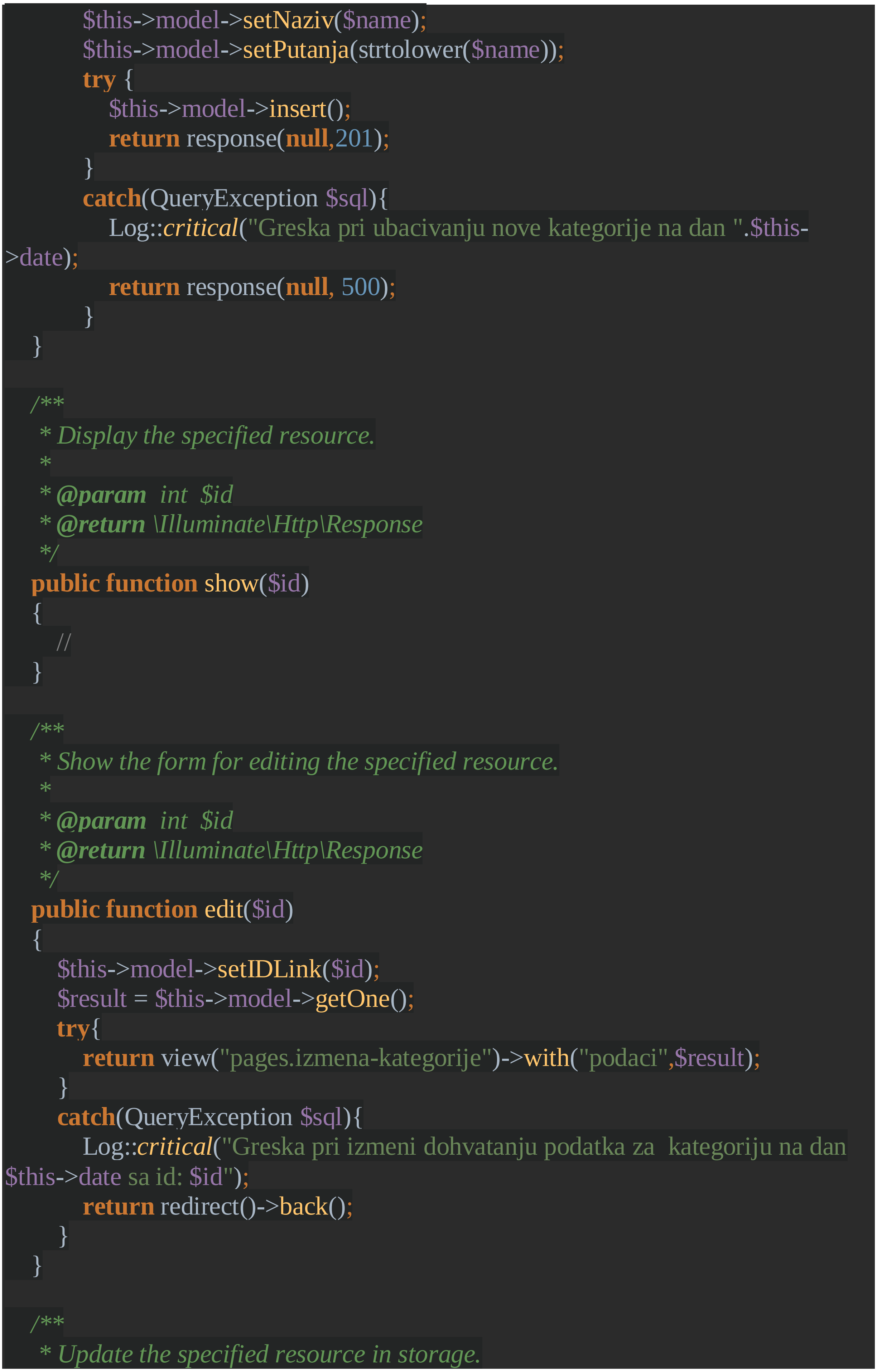
ActivityController.php

```
<?php

namespace App\Http\Controllers\Admin;

use App\Models\Izvestaj;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;

class ActivityController extends Controller
{
    public function index(){
        $m = new Izvestaj();
        $a['data'] =  $m->getAll();
        return view('pages.izvestaj', $a);
    }
}
```

GenreController.php

```
<?php

namespace App\Http\Controllers\Admin;
```

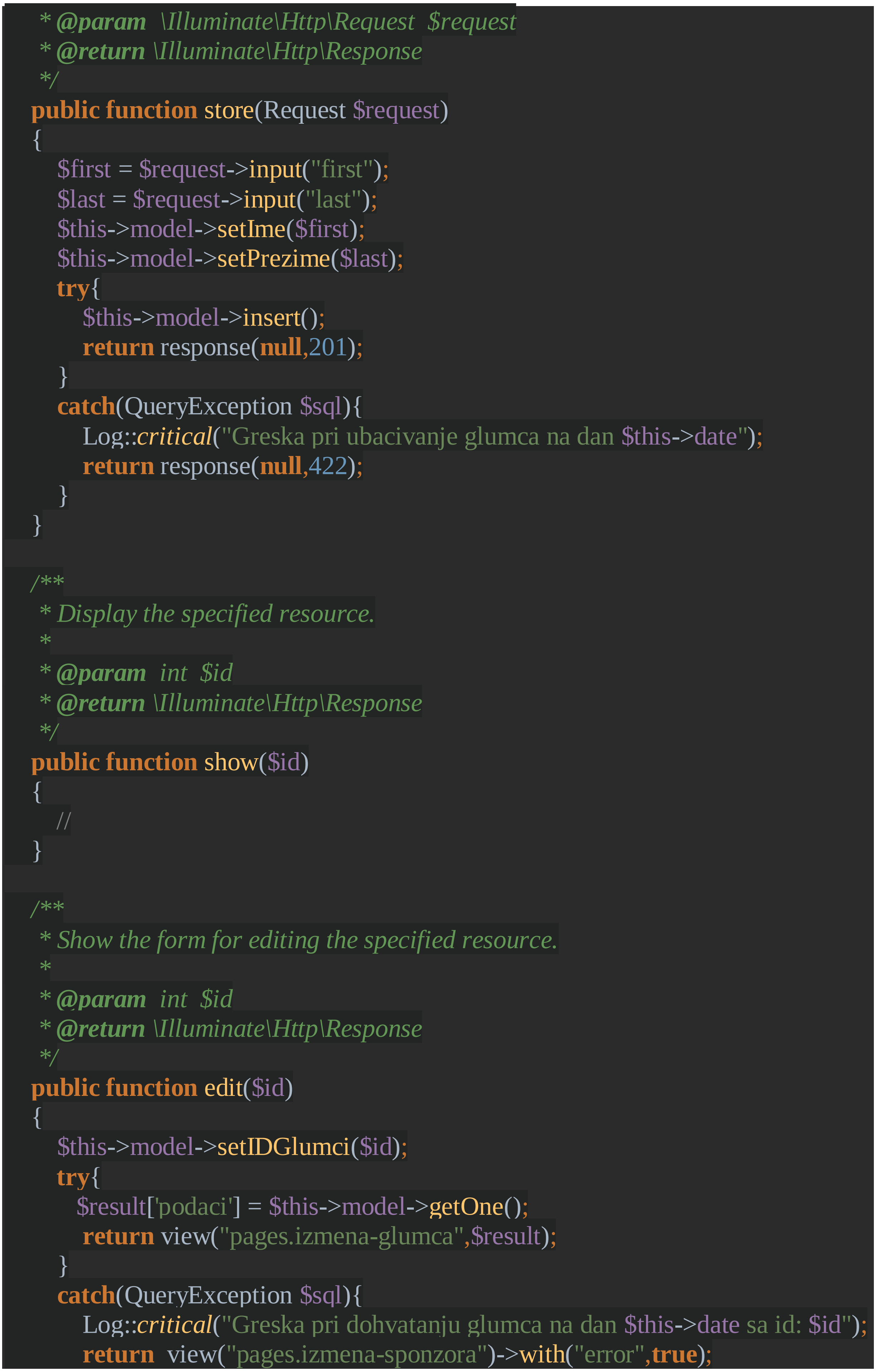
```php
use App\Http\Controllers\Controller;
use App\Http\Requests\GenreRequest;
use App\Models\Link;
use Illuminate\Database\QueryException;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Log;

class GenreController extends Controller
{
    private $model;
    private $date;
    public function __construct(){
        $this->model = new Link();
        $this->date = date("d.m.Y H:i:s");
    }

    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $data['kategorije'] = $this->model->getCategories();
        return view("pages.kategorije",$data);
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        return view("pages.dodaj-kategoriju");
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @return \Illuminate\Http\Response
     */
    public function store(GenreRequest $request)
    {
            $name = $request->input("category");
```



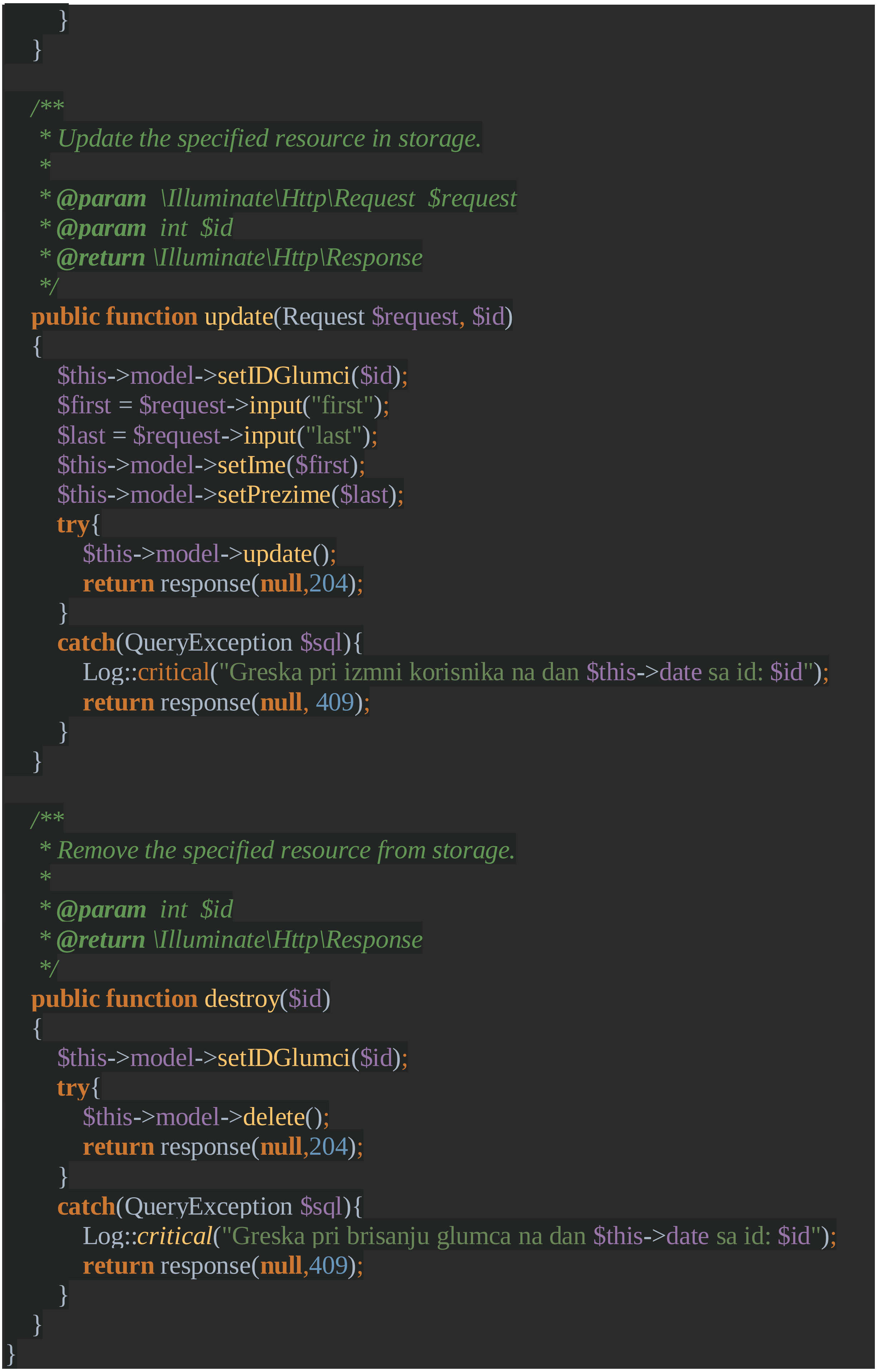
```
            $this->model->setNaziv($name);
            $this->model->setPutanja(strtolower($name));
            try {
                $this->model->insert();
                return response(null,201);
            }
            catch(QueryException $sql){
                Log::critical("Greska pri ubacivanju nove kategorije na dan ".$this-
>date);
                return response(null, 500);
            }
    }

    /**
     * Display the specified resource.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        //
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function edit($id)
    {
        $this->model->setIDLink($id);
        $result = $this->model->getOne();
        try{
            return view("pages.izmena-kategorije")->with("podaci",$result);
        }
        catch(QueryException $sql){
            Log::critical("Greska pri izmeni dohvatanju podatka za  kategoriju na dan
$this->date sa id: $id");
            return redirect()->back();
        }
    }

    /**
     * Update the specified resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function update(GenreRequest $request, $id)
    {
        $name = $request->input("category");
        $this->model->setIDLink($id);
        $this->model->setNaziv($name);
        $this->model->setPutanja(strtolower($name));

        try{
            $this->model->update();
            return response(null, 204);
        }
        catch(QueryException $sql){
            Log::critical("Greska pri izmeni kategorije na dan $this->date za kategoriju sa id: $id");
            return response(null,500);
        }
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function destroy($id)
    {
        $this->model->setIDLink($id);
        try{
            $this->model->delete();
            return response(null,204);
        }
        catch(QueryException $sql){
            Log::critical("Greska pri brisanju kategorije na dan $this->date za kategoiju sa id: $id");
            return response(null,500);
        }
    }
}
```

RoleController.php

```php
<?php

namespace App\Http\Controllers\Admin;

use App\Models\Glumci;
use Illuminate\Database\QueryException;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Log;

class RoleController extends Controller
{
    private $model;
    private $date;
    public function __construct()
    {
        $this->model = new Glumci();
        $this->date = date("d.m.Y H:i:s");
    }

    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $result['glumci'] =  $this->model->getAll();
        return view("pages.glumci" ,$result);
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        return view("pages.dodaj-glumca");
    }

    /**
     * Store a newly created resource in storage.
     *
```



```
     * @param  \Illuminate\Http\Request  $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
        $first = $request->input("first");
        $last = $request->input("last");
        $this->model->setIme($first);
        $this->model->setPrezime($last);
        try{
            $this->model->insert();
            return response(null,201);
        }
        catch(QueryException $sql){
            Log::critical("Greska pri ubacivanje glumca na dan $this->date");
            return response(null,422);
        }
    }

    /**
     * Display the specified resource.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        //
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function edit($id)
    {
        $this->model->setIDGlumci($id);
        try{
           $result['podaci'] = $this->model->getOne();
            return view("pages.izmena-glumca",$result);
        }
        catch(QueryException $sql){
            Log::critical("Greska pri dohvatanju glumca na dan $this->date sa id: $id");
            return  view("pages.izmena-sponzora")->with("error",true);
```

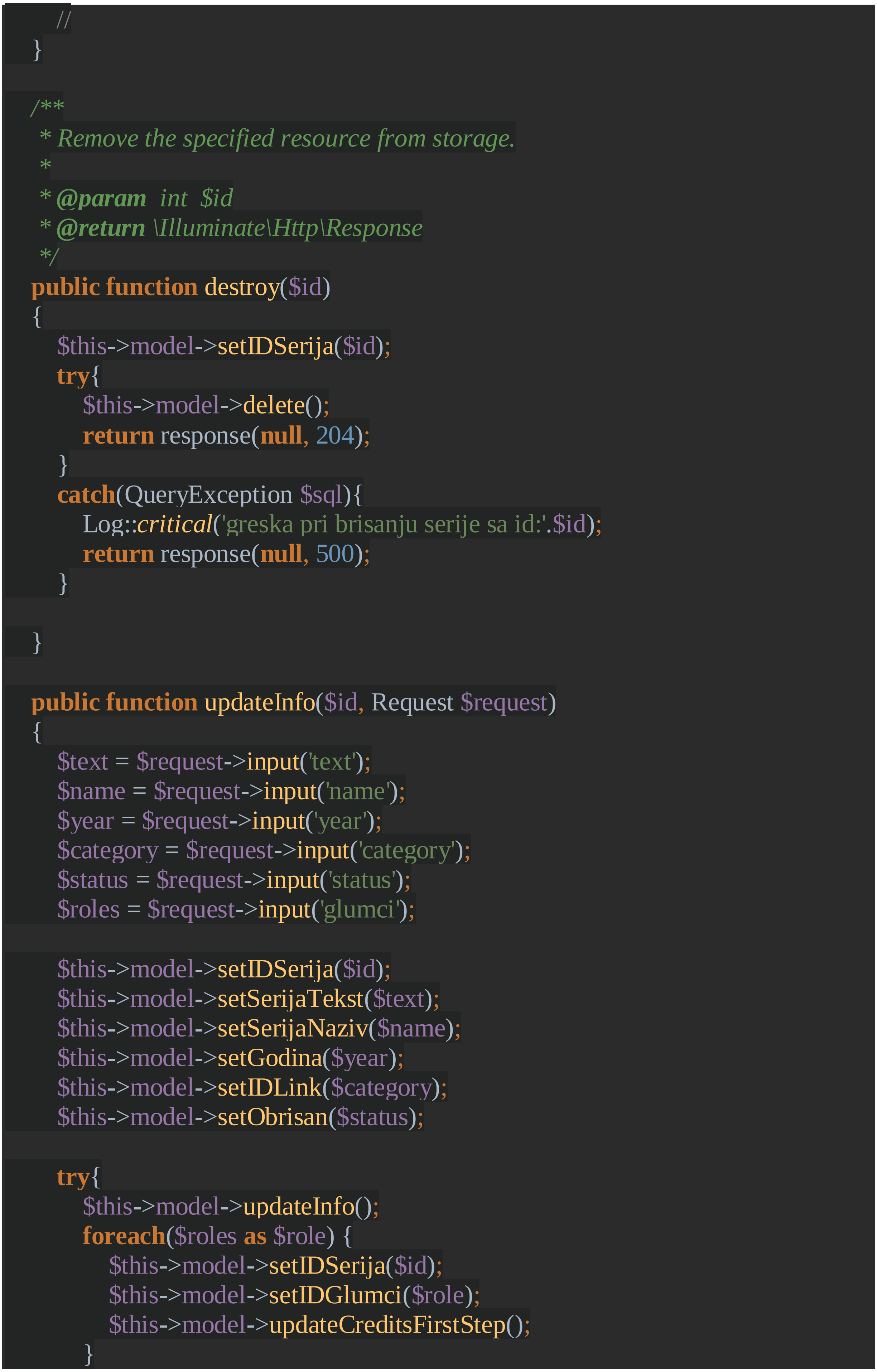


```
        }
    }

    /**
     * Update the specified resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request, $id)
    {
        $this->model->setIDGlumci($id);
        $first = $request->input("first");
        $last = $request->input("last");
        $this->model->setIme($first);
        $this->model->setPrezime($last);
        try{
            $this->model->update();
            return response(null,204);
        }
        catch(QueryException $sql){
            Log::critical("Greska pri izmni korisnika na dan $this->date sa id: $id");
            return response(null, 409);
        }
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function destroy($id)
    {
        $this->model->setIDGlumci($id);
        try{
            $this->model->delete();
            return response(null,204);
        }
        catch(QueryException $sql){
            Log::critical("Greska pri brisanju glumca na dan $this->date sa id: $id");
            return response(null,409);
        }
    }
}
```

SeriesController.php

```
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Requests\AdminInsertSeriesRequest;
use App\Http\Requests\PictureRequest;
use App\Models\Glumci;
use App\Models\Link;
use App\Models\Serija;
use Illuminate\Database\QueryException;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Log;

class SeriesController extends Controller
{
    private $model;
    public function __construct()
    {
        $this->model = new Serija();
    }

    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        $podaci = $this->model->getAll();
        return view('pages.serije-spisak')->with('serije', $podaci);
    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        $link = new Link();
        $m = new Glumci();
        $cat = $link->getCategories();
        $roles = $m->getAll();
        return view('pages.dodaj-seriju')
            ->with('kategorije' , $cat)
            ->with('glumci', $roles);
    }


    public function store(AdminInsertSeriesRequest $request)
    {
        $name = $request->input('name');
        $year = $request->input('year');
        $cat = $request->input('category');
        $roles = $request->input('glumci');


        $text = $request->input('text');
        $pic = $request->file('img');
        $ime = "images/".time()."_".$pic->getClientOriginalName();

        $this->model->setSerijaNaziv($name);
        $this->model->setSerijaTekst($text);
        $this->model->setGodina($year);
        $this->model->setIDLink($cat);
        $this->model->setSlika($ime);

        try {
            if($pic->isValid()) {
                $pic->move(public_path("images/"),  $ime);
                $id = $this->model->insert();
                foreach($roles as $role) {
                    $this->model->setIDGlumci($role);
                    $this->model->setIDSerija($id);
                    $this->model->insertCredits();
                }
                return redirect()->back()->with("uspeh", "Uspešno ste dodali
sponzora");
            }
        }
        catch(QueryException $sql){
            Log::critical("Greska pri ubacivanju serije u bazu!");
            return redirect()->back()->with("error", "Doslo je do greske, kontaktirajte
admina!");
        }

    }

    /**
```

```
     * Display the specified resource.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        //
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function edit($id, $cat)
    {
        $this->model->setIDSerija($id);
        $link = new Link();
        $link->setIDLink($cat);
        try{
            $podaci['serija'] = $this->model->getOne();
            $glumac = new Glumci();
            $podaci['svi_glumci'] = $glumac->getAll();
            $podaci['glumci'] =  $this->model->getRoles();

            $podaci['kategorije'] = $link->getOtherCategoreis();
            return view('pages.izmena-serije', $podaci);
        }
        catch(QueryException $sql){
            Log::critical('Greška pri dohvatanju informacija za seriju sa id: '.$id);
            return redirect()->back()->with("error", "Doslo je do greske, kontaktirajte
admina!");
        }
    }

    /**
     * Update the specified resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request, $id)
    {
```



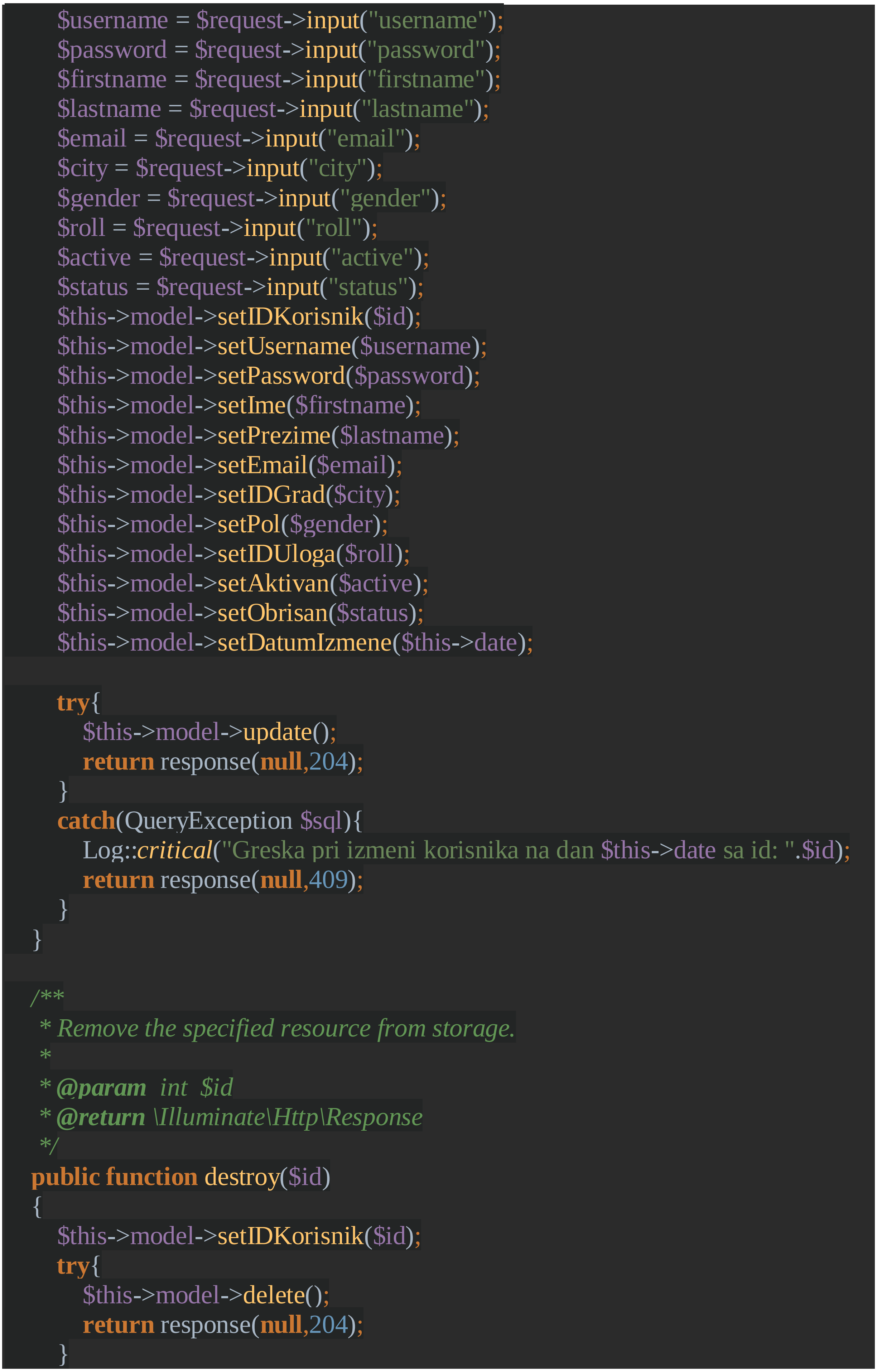
```
        //
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function destroy($id)
    {
        $this->model->setIDSerija($id);
        try{
            $this->model->delete();
            return response(null, 204);
        }
        catch(QueryException $sql){
            Log::critical('greska pri brisanju serije sa id:'.$id);
            return response(null, 500);
        }

    }

    public function updateInfo($id, Request $request)
    {
        $text = $request->input('text');
        $name = $request->input('name');
        $year = $request->input('year');
        $category = $request->input('category');
        $status = $request->input('status');
        $roles = $request->input('glumci');

        $this->model->setIDSerija($id);
        $this->model->setSerijaTekst($text);
        $this->model->setSerijaNaziv($name);
        $this->model->setGodina($year);
        $this->model->setIDLink($category);
        $this->model->setObrisan($status);

        try{
            $this->model->updateInfo();
            foreach($roles as $role) {
                $this->model->setIDSerija($id);
                $this->model->setIDGlumci($role);
                $this->model->updateCreditsFirstStep();
            }
            foreach($roles as $role) {
                $this->model->setIDSerija($id);
                $this->model->setIDGlumci($role);
                $this->model->updateCreditsFinalStep();
            }
        return redirect(route('series-show'))->with('uspeh', "Uspešno ste izmenili
seriju");
        }
        catch(QueryException $sql){
            Log::critical('Greska pri izmeni serije sa id'.$id.$sql->getMessage());
            return redirect(route('series-show'))->with('error', "Došlo je od greške
konaktirajte admina");
        }

    }
    public function updatePicture($id, PictureRequest $picture)
    {
            $this->model->setIDSerija($id);
            $image = $picture->file('img');
            $ime = "images/".time().'_'.$image->getClientOriginalName();
            public_path("images/");
            if($image->isValid()){
                $image->move(public_path('images/'),$ime);
                $this->model->setSlika($ime);
                try{
                    $this->model->updatePicture();
                    return redirect(route('series-show'))->with('uspeh', "Uspešno ste
izmenili sliku ");
                }
                catch(QueryException $sql){
                    Log::critical('Greska pri izmeni slike za seriju sa id:'.$id);
                    return redirect(route('series-show'))->with('error', "Došlo je od greške
konaktirajte admina");
                }
            }
    }
}
```

SponsorController.php

```php
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Requests\PictureRequest;
use App\Http\Requests\SponsorRequest;
use Illuminate\Database\QueryException;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use \App\Models\Sponzor;
use Illuminate\Support\Facades\Log;

class SponsorController extends Controller
{
    private $model;
    private $date;
    public function __construct()
    {
        $this->model = new Sponzor();
        $this->date = date("d.m.Y H:i:s");
    }

    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
        try{
            $result = $this->model->getSponsors();
            return view("pages.sponzori")->with("sponzori", $result);
        }
        catch(QueryException $sql){
            Log::critical("Greska pri ucitavanju sponzora");
            return redirect()->back();
        }

    }

    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        return view('pages.dodaj-sponzora');
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @return \Illuminate\Http\Response
     */
    public function store(SponsorRequest $request, PictureRequest $pictureRequest)
    {
        $link = $request->input('link');
        $slika = $pictureRequest->file("img");
        $ime = "images/".time()."_".$slika->getClientOriginalName();
        $this->model->setLink($link);
        $this->model->setSlika($ime);
        try {
            if($slika->isValid()) {
                $slika->move(public_path("images/"),  $ime);
                $this->model->insertSponsor();
                return redirect()->back()->with("uspeh", "Uspešno ste dodali sponzora");
            }
            else{
                    return redirect()->back();
                }
            }
        catch(QueryException $sql){
                Log::critical("Greska pri ubacivanju sponzora u bazu!".$this->date);
                return redirect()->back()->with("error", "Doslo je do greske, kontaktirajte admina!");
        }
    }

    /**
     * Display the specified resource.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        //
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function edit($id)
    {
        $this->model->setIDSponsor($id);

        try {
            $result['podaci'] = $this->model->getOneSponsor();
            return view("pages.izmena-sponzora",$result);
        }
        catch(QueryException $sql){
            Log::critical("Greska pri dohvatanju jednog sponzora za izmenu". $this-
>date);
            return  view("pages.izmena-sponzora")->with("error",true);
        }
    }

    /**
     * Update the specified resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function updateLink(SponsorRequest $request, $id)
    {
        $link = $request->input('link');
        $this->model->setLink($link);
        $this->model->setIDSponsor($id);
        try {
            $this->model->updateLink();
            return redirect(route("sponsor-show"))->with("uspeh",true);
        }
        catch(QueryException $sql){
            Log::critical("Greska pri izmeni linka za sponzora". $this->date);
            return  redirect(route("sponsor-show"))->with("error", true);
        }

    }
    public function updatePicture(PictureRequest $pictureRequest, $id){

        $picture = $pictureRequest->file("img");
        $name = time()."_".$picture->getClientOriginalName();
        $this->model->setSlika($name);
        $this->model->setIDSponsor($id);
        $p =  public_path("\images");

        try{
            if($picture->isValid()){
                dd($p);
                $picture->move(public_path("\images"),$name);
                $this->model->updatePicture();
                return redirect(route("sponsor-show"))->with("uspeh", true);
            }
        }
        catch(QueryException $sql){
            Log::critical("Greska pri izmeni slike za sponzora". $this->date);
            return  redirect(route("sponsor-show"))->with("error",true);
        }
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function destroy($id)
    {
        $this->model->setIDSponsor($id);
        try{
            $this->model->deleteSponsor();
            return response("Uspešno ste obrisali sponzora",204);
        }
        catch(QueryException $sql){
            Log::critical("Greska pri brisanju sponzora sa id: ".$id. " " . $this->date);
            return response("Greska pri brisanju sponzora", 409);
        }
    }
}
```

UserController.php

```php
<?php

namespace App\Http\Controllers\Admin;

use App\Http\Requests\AdminInsertUserRequest;
use App\Http\Requests\LogRequest;
use App\Http\Requests\PictureRequest;
use App\Models\Grad;
use App\Models\Korisnik;
use App\Models\Uloga;
use Illuminate\Database\QueryException;
use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Log;

class UserController extends Controller
{
    private $model;
    private $date;
    private $result;
    public function __construct()
    {
        $this->model = new Korisnik();
        $this->date = date("d.m.Y H:i:s");
        $city = new Grad();
        $roles = new Uloga();
        $this->result['uloge'] = $roles->getRoles();
        $this->result['gradovi'] = $city->getAll();
    }

    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function index()
    {
            try{
                $result = $this->model->getUsers();
                return view("pages.korisnici")->with("korisnici", $result);
            }
            catch(QueryException $sql){
                Log::critical("Greska pri ucitavanju sponzora ".$this->date);
                return redirect()->back();
        }
    }
    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        return view("pages.dodaj-korisnika",$this->result);
    }

    /**
     * Store a newly created resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @return \Illuminate\Http\Response
     */
    public function store(AdminInsertUserRequest $request)
    {
        $username = $request->input("username");
        $password = $request->input("password");
        $firstname = $request->input("firstname");
        $lastname = $request->input("lastname");
        $email = $request->input("email");
        $city = $request->input("city");
        $gender = $request->input("gender");
        $roll = $request->input("roll");
        $active = $request->input("active");
        $token = md5($username.time());
        $this->model->setUsername($username);
        $this->model->setPassword($password);
        $this->model->setIme($firstname);
        $this->model->setPrezime($lastname);
        $this->model->setEmail($email);
        $this->model->setIDGrad($city);
        $this->model->setPol($gender);
        $this->model->setIDUloga($roll);
        $this->model->setAktivan($active);
        $this->model->setDatumRegistracije($this->date);
        $this->model->setToken($token);
        try{
            $this->model->insert();
            return response(null,201);
        }
        catch(QueryException $sql){
            Log::critical("Greska pri upisivanju korisnika od strane admina! ".$this-
>date);
            return response(null, 409);
        }
    }

    /**
     * Display the specified resource.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        //
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function edit($id)
    {
        $this->model->setIDKorisnik($id);
        try{
            $podaci = $this->model->getOneUser();
            return view("pages.izmeni-korisnika",$this->result)->with("podaci",
$podaci);
        }
        catch(QueryException $sql){
            Log::critical("Greska pri dohvatnju jednog korisnika ".$this->date);
            return redirect(route('user-show'))->with("error",true);
        }
    }

    /**
     * Update the specified resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function update(Request $request, $id)
    {
```



```
        $username = $request->input("username");
        $password = $request->input("password");
        $firstname = $request->input("firstname");
        $lastname = $request->input("lastname");
        $email = $request->input("email");
        $city = $request->input("city");
        $gender = $request->input("gender");
        $roll = $request->input("roll");
        $active = $request->input("active");
        $status = $request->input("status");
        $this->model->setIDKorisnik($id);
        $this->model->setUsername($username);
        $this->model->setPassword($password);
        $this->model->setIme($firstname);
        $this->model->setPrezime($lastname);
        $this->model->setEmail($email);
        $this->model->setIDGrad($city);
        $this->model->setPol($gender);
        $this->model->setIDUloga($roll);
        $this->model->setAktivan($active);
        $this->model->setObrisan($status);
        $this->model->setDatumIzmene($this->date);

        try{
            $this->model->update();
            return response(null,204);
        }
        catch(QueryException $sql){
            Log::critical("Greska pri izmeni korisnika na dan $this->date sa id: ".$id);
            return response(null,409);
        }
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param  int  $id
     * @return \Illuminate\Http\Response
     */
    public function destroy($id)
    {
        $this->model->setIDKorisnik($id);
        try{
            $this->model->delete();
            return response(null,204);
        }
        catch(QueryException $sql){
            Log::critical("Greska pri brisanju korisnika na dan $this->date sa id: $id");
            return response(null,500);
        }
    }
}
```

-----------------------------------Models---

Glumci.php

```
<?php
/**
 * Created by PhpStorm.
 * User: Dušan
 * Date: 8.3.2019.
 * Time: 16.42
 */

namespace App\Models;


use Illuminate\Support\Facades\DB;

class Glumci
{
    private $ID_glumci;
    private $ime;
    private $prezime;

    /**
     * @return mixed
     */
    public function getIDGlumci()
    {
        return $this->ID_glumci;
    }

    /**
     * @param mixed $ID_glumci
     */
    public function setIDGlumci($ID_glumci): void
    {
        $this->ID_glumci = $ID_glumci;
    }

    /**
     * @return mixed
     */
    public function getIme()
    {
        return $this->ime;
    }

    /**
     * @param mixed $ime
     */
    public function setIme($ime): void
    {
        $this->ime = $ime;
    }

    /**
     * @return mixed
     */
    public function getPrezime()
    {
        return $this->prezime;
    }

    /**
     * @param mixed $prezime
     */
    public function setPrezime($prezime): void
    {
        $this->prezime = $prezime;
    }
    private $table = "glumci";

    public function getAll(){
        return DB::table($this->table)->get();
    }
    public function delete(){
        DB::transaction(function (){
            DB::table("serija_glumci")->where("ID_glumci",$this->ID_glumci)-
>delete();
            DB::table($this->table)->where("ID_glumci", $this->ID_glumci)->delete();
        });
    }
```

```php
    public function update(){
        DB::table($this->table)->where("ID_glumci", $this->ID_glumci)->update([
            "glumci_ime" => $this->ime,
            "glumci_prezime" => $this->prezime
            ]);
    }
    public function insert(){
        DB::table($this->table)->insert([
            "glumci_ime" => $this->ime,
            "glumci_prezime" => $this->prezime
        ]);
    }
    public function getOne(){
       return DB::table($this->table)->where("ID_glumci", $this->ID_glumci)->get();
    }


}
```

Grad.php

```php
<?php
/**
 * Created by PhpStorm.
 * User: Dušan
 * Date: 25.2.2019.
 * Time: 00.18
 */

namespace App\Models;


use Illuminate\Support\Facades\DB;

class Grad
{
    private $ID_grad;
    private $grad_ime;

    /**
     * @return mixed
     */
    public function getIDGrad()
    {
        return $this->ID_grad;
    }

    /**
     * @param mixed $ID_grad
     */
    public function setIDGrad($ID_grad): void
    {
        $this->ID_grad = $ID_grad;
    }

    /**
     * @return mixed
     */
    public function getGradIme()
    {
        return $this->grad_ime;
    }

    /**
     * @param mixed $grad_ime
     */
    public function setGradIme($grad_ime): void
    {
        $this->grad_ime = $grad_ime;
    }

    public function getAll(){
        $query = DB::table("grad")->get();
        return $query;
    }
}
```

Izvestaj.php

```
<?php
/**
 * Created by PhpStorm.
 * User: ASUS
 * Date: 12.3.2019
 * Time: 0:59
 */

namespace App\Models;
```

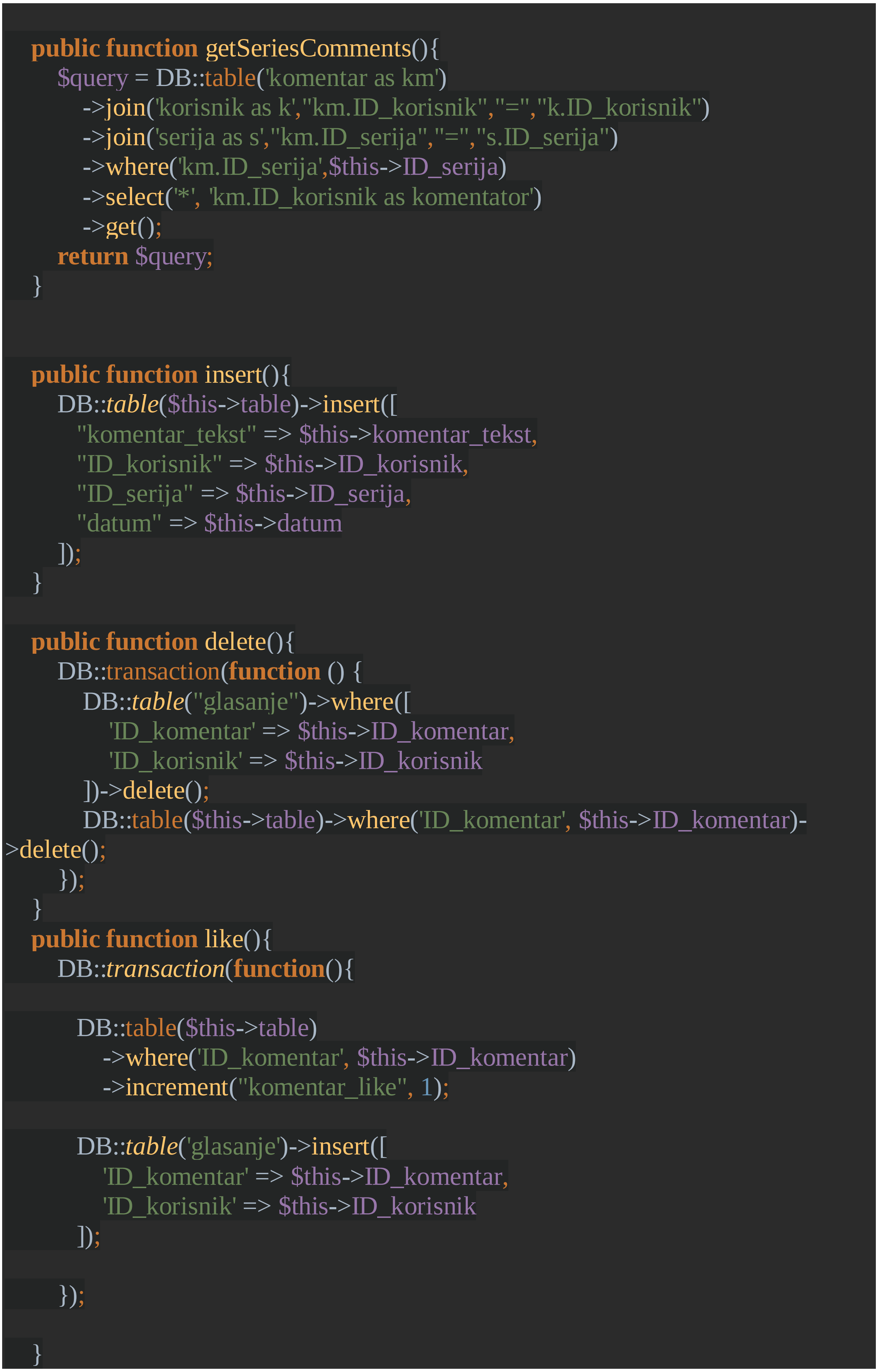
```
use Illuminate\Support\Facades\DB;

class Izvestaj
{
    /**
     * @param mixed $ID_korisnik
     */
    public function setIDKorisnik($ID_korisnik): void
    {
        $this->ID_korisnik = $ID_korisnik;
    }
    private $ID_korisnik;
    private $datum;

    private $table = 'izvestaj';
    /**
     * @param mixed $datum
     */
    public function setDatum($datum): void
    {
        $this->datum = $datum;
    }

    public function insert($activity) {
        DB::table($this->table)->insert([
            'ID_korisnik' => $this->ID_korisnik,
            'datum' => $this->datum,
            'aktivnost' => $activity
        ]);
    }

    public function getAll(){
        return DB::table('izvestaj as i')
            ->join('korisnik as k','i.ID_korisnik', '=','k.ID_korisnik')
            ->get();
    }

}
```

Komentar.php

```
<?php
/**
 * Created by PhpStorm.
 * User: Dušan
 * Date: 4.3.2019.
 * Time: 22.46
 */

namespace App\Models;


use function foo\func;
use Illuminate\Support\Facades\DB;

class Komentar
{
    private $ID_komentar;
    private $komentar_tekst;
    private $ID_korisnik;
    private $ID_serija;
    private $datum;
    private $komentar_like;
    private $komentar_dislike;
    private $table = "komentar";

    /**
     * @return mixed
     */
    public function getIDKomentar()
    {
        return $this->ID_komentar;
    }

    /**
     * @param mixed $ID_komentar
     */
    public function setIDKomentar($ID_komentar): void
    {
        $this->ID_komentar = $ID_komentar;
    }

    /**
     * @return mixed
     */
```

```php
public function getKomentarTekst()
{
    return $this->komentar_tekst;
}

/**
 * @param mixed $komentar_tekst
 */
public function setKomentarTekst($komentar_tekst): void
{
    $this->komentar_tekst = $komentar_tekst;
}

/**
 * @return mixed
 */
public function getIDKorisnik()
{
    return $this->ID_korisnik;
}

/**
 * @param mixed $ID_korisnik
 */
public function setIDKorisnik($ID_korisnik): void
{
    $this->ID_korisnik = $ID_korisnik;
}

/**
 * @return mixed
 */
public function getIDSerija()
{
    return $this->ID_serija;
}

/**
 * @param mixed $ID_serija
 */
public function setIDSerija($ID_serija): void
{
    $this->ID_serija = $ID_serija;
}

/**
```

```
     * @return mixed
     */
    public function getDatum()
    {
        return $this->datum;
    }

    /**
     * @param mixed $datum
     */
    public function setDatum($datum): void
    {
        $this->datum = $datum;
    }

    /**
     * @return mixed
     */
    public function getKomentarLike()
    {
        return $this->komentar_like;
    }

    /**
     * @param mixed $komentar_like
     */
    public function setKomentarLike($komentar_like): void
    {
        $this->komentar_like = $komentar_like;
    }

    /**
     * @return mixed
     */
    public function getKomentarDislike()
    {
        return $this->komentar_dislike;
    }

    /**
     * @param mixed $komentar_dislike
     */
    public function setKomentarDislike($komentar_dislike): void
    {
        $this->komentar_dislike = $komentar_dislike;
    }
```

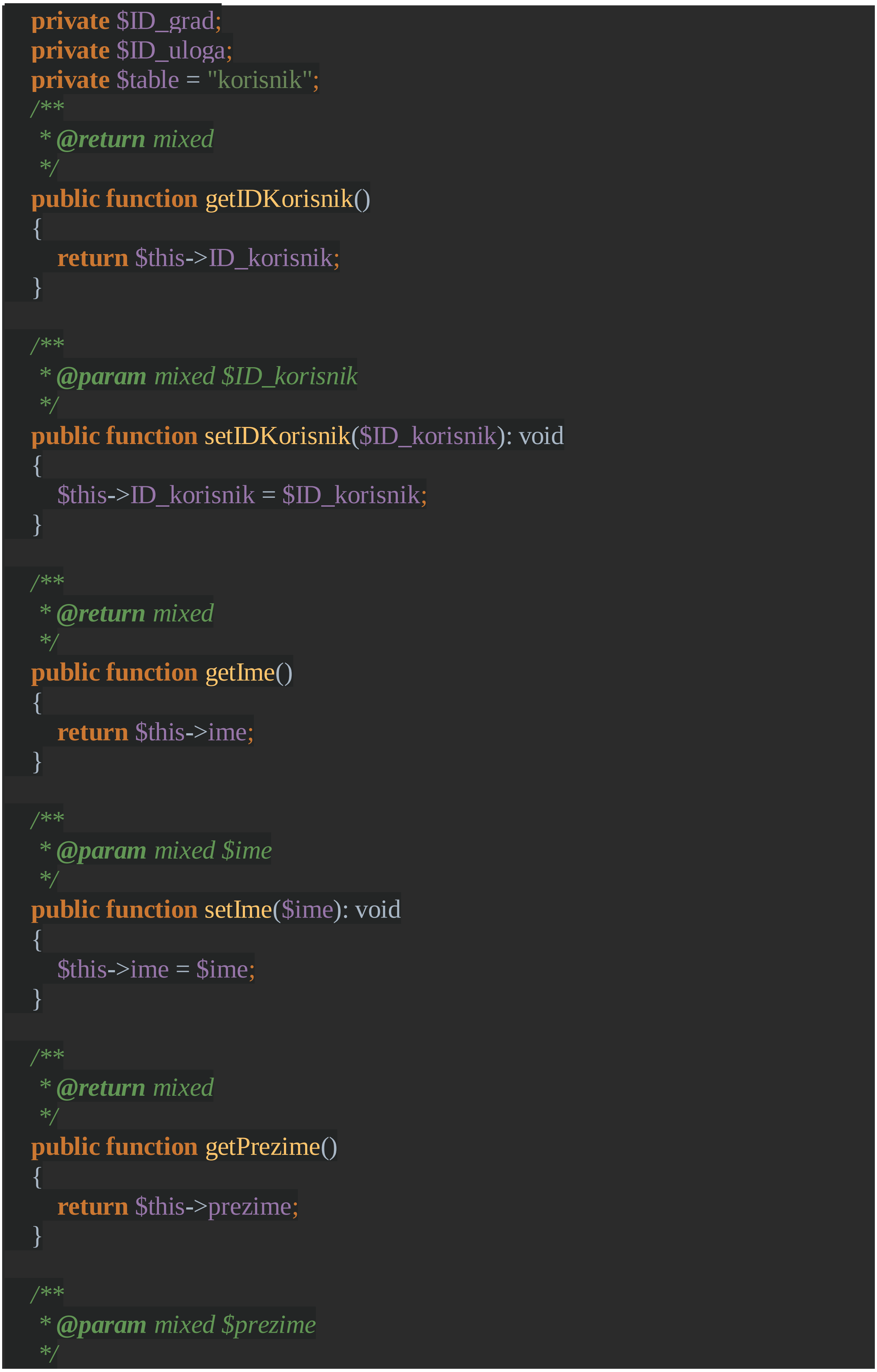


```
  public function getSeriesComments(){
     $query = DB::table('komentar as km')
        ->join('korisnik as k',"km.ID_korisnik","=","k.ID_korisnik")
        ->join('serija as s',"km.ID_serija","=","s.ID_serija")
        ->where('km.ID_serija',$this->ID_serija)
        ->select('*', 'km.ID_korisnik as komentator')
        ->get();
     return $query;
  }


  public function insert(){
     DB::table($this->table)->insert([
       "komentar_tekst" => $this->komentar_tekst,
       "ID_korisnik" => $this->ID_korisnik,
       "ID_serija" => $this->ID_serija,
       "datum" => $this->datum
     ]);
  }

  public function delete(){
     DB::transaction(function () {
        DB::table("glasanje")->where([
           'ID_komentar' => $this->ID_komentar,
           'ID_korisnik' => $this->ID_korisnik
        ])->delete();
        DB::table($this->table)->where('ID_komentar', $this->ID_komentar)-
>delete();
     });
  }
  public function like(){
     DB::transaction(function(){

       DB::table($this->table)
          ->where('ID_komentar', $this->ID_komentar)
          ->increment("komentar_like", 1);

       DB::table('glasanje')->insert([
          'ID_komentar' => $this->ID_komentar,
          'ID_korisnik' => $this->ID_korisnik
       ]);

     });

  }
```

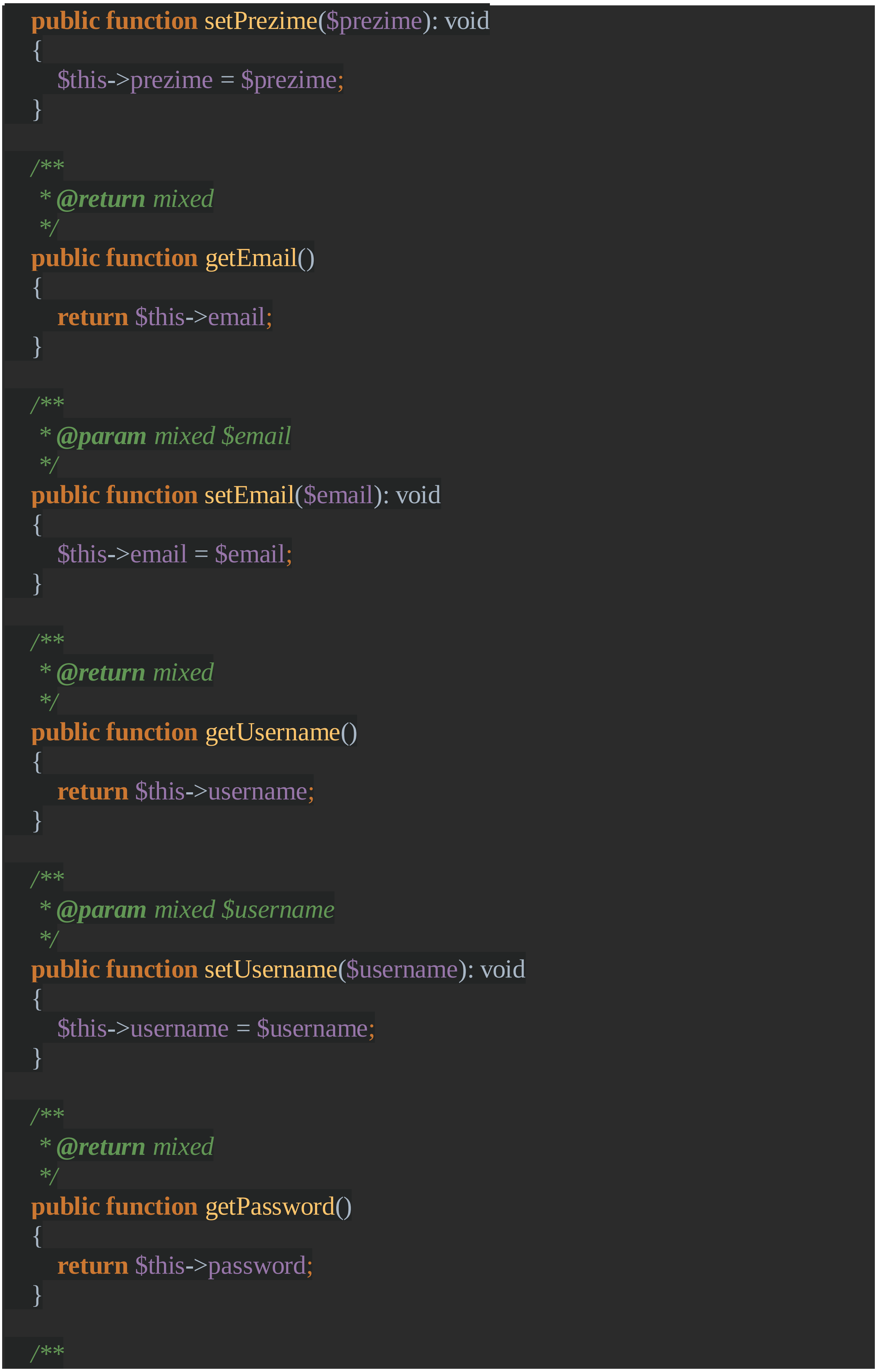
```
    public function dislike(){
        DB::transaction(function(){

            DB::table($this->table)
                ->where('ID_komentar', $this->ID_komentar)
                ->increment("komentar_dislike", 1);

            DB::table('glasanje')->insert([
                'ID_komentar' => $this->ID_komentar,
                'ID_korisnik' => $this->ID_korisnik
            ]);

        });

    }

}
```

Korisnik.php

```
<?php
/**
 * Created by PhpStorm.
 * User: Dušan
 * Date: 23.2.2019.
 * Time: 15.10
 */

namespace App\Models;
use Illuminate\Support\Facades\DB;

class Korisnik
{
    private $ID_korisnik;
    private $ime;
    private $prezime;
    private $email;
    private $username;
    private $password;
    private $token;
    private $pol;
    private $aktivan;
    private $datum_registracije;
    private $datum_izmene;
    private $obrisan;
```

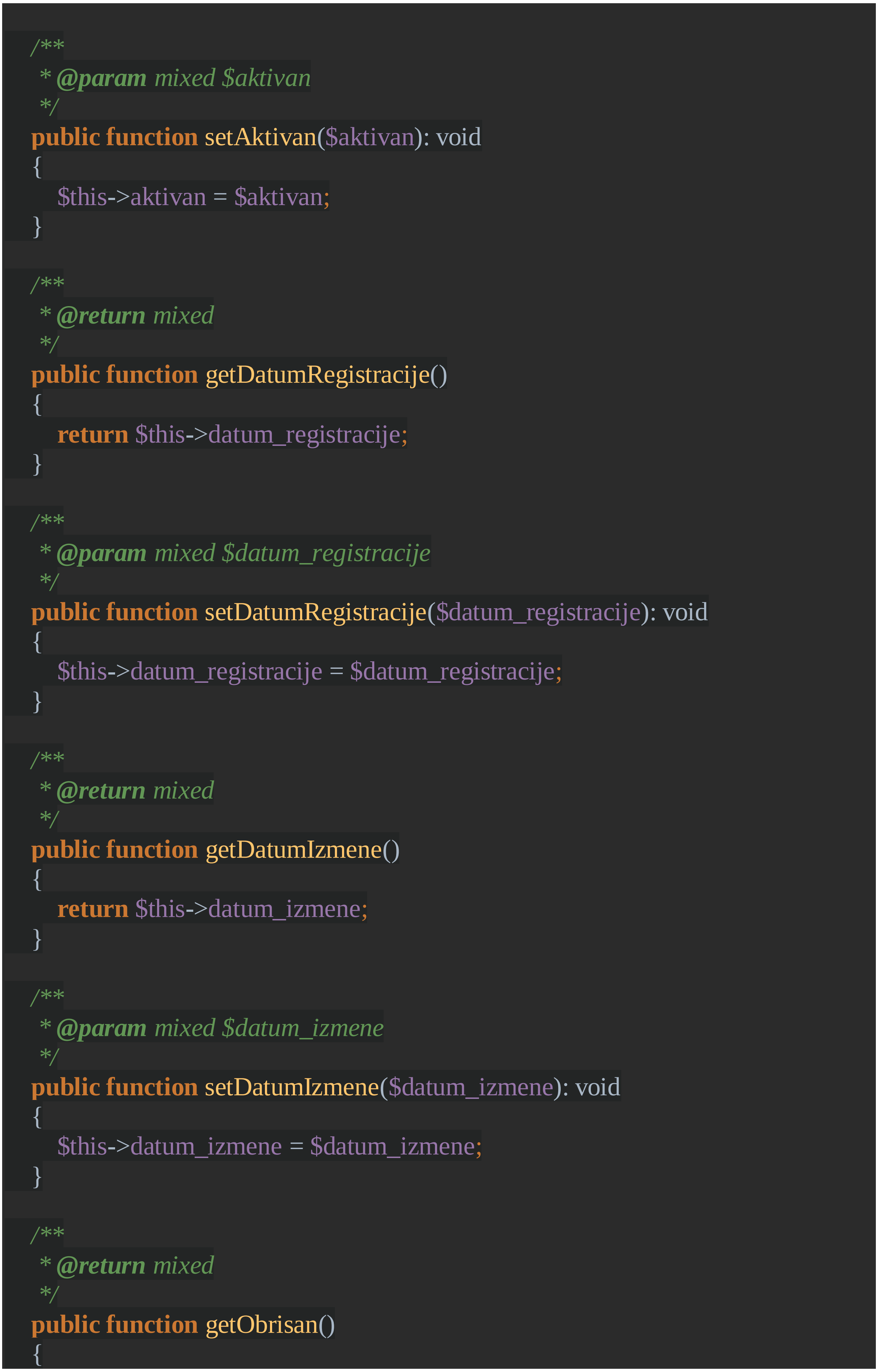


```
    private $ID_grad;
    private $ID_uloga;
    private $table = "korisnik";
    /**
     * @return mixed
     */
    public function getIDKorisnik()
    {
        return $this->ID_korisnik;
    }

    /**
     * @param mixed $ID_korisnik
     */
    public function setIDKorisnik($ID_korisnik): void
    {
        $this->ID_korisnik = $ID_korisnik;
    }

    /**
     * @return mixed
     */
    public function getIme()
    {
        return $this->ime;
    }

    /**
     * @param mixed $ime
     */
    public function setIme($ime): void
    {
        $this->ime = $ime;
    }

    /**
     * @return mixed
     */
    public function getPrezime()
    {
        return $this->prezime;
    }

    /**
     * @param mixed $prezime
     */
```

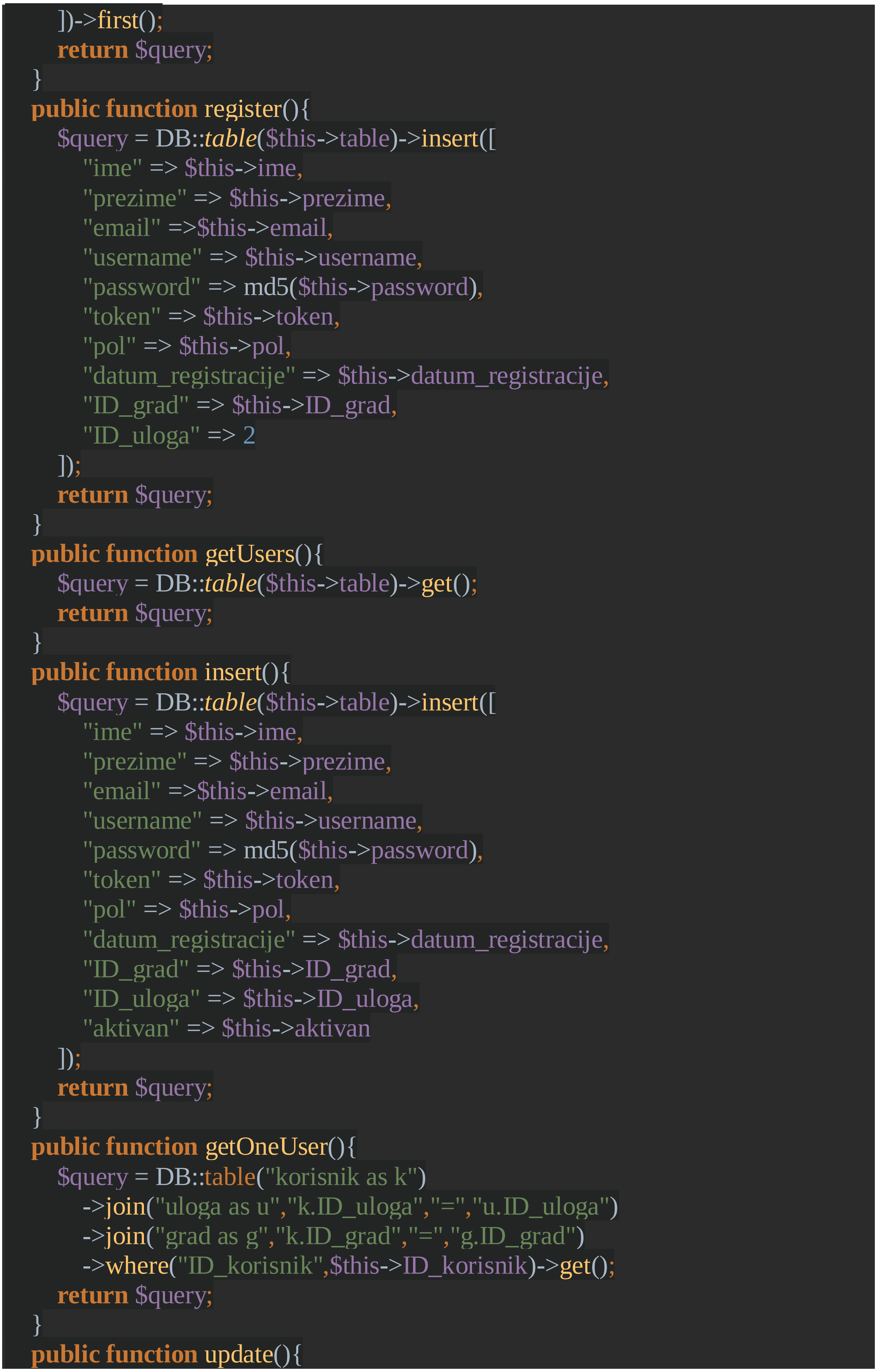


```php
    public function setPrezime($prezime): void
    {
        $this->prezime = $prezime;
    }

    /**
     * @return mixed
     */
    public function getEmail()
    {
        return $this->email;
    }

    /**
     * @param mixed $email
     */
    public function setEmail($email): void
    {
        $this->email = $email;
    }

    /**
     * @return mixed
     */
    public function getUsername()
    {
        return $this->username;
    }

    /**
     * @param mixed $username
     */
    public function setUsername($username): void
    {
        $this->username = $username;
    }

    /**
     * @return mixed
     */
    public function getPassword()
    {
        return $this->password;
    }

    /**
```

```
     * @param mixed $password
     */
    public function setPassword($password): void
    {
        $this->password = $password;
    }

    /**
     * @return mixed
     */
    public function getToken()
    {
        return $this->token;
    }

    /**
     * @param mixed $token
     */
    public function setToken($token): void
    {
        $this->token = $token;
    }

    /**
     * @return mixed
     */
    public function getPol()
    {
        return $this->pol;
    }

    /**
     * @param mixed $pol
     */
    public function setPol($pol): void
    {
        $this->pol = $pol;
    }

    /**
     * @return mixed
     */
    public function getAktivan()
    {
        return $this->aktivan;
    }
```



```
/**
 * @param mixed $aktivan
 */
public function setAktivan($aktivan): void
{
    $this->aktivan = $aktivan;
}

/**
 * @return mixed
 */
public function getDatumRegistracije()
{
    return $this->datum_registracije;
}

/**
 * @param mixed $datum_registracije
 */
public function setDatumRegistracije($datum_registracije): void
{
    $this->datum_registracije = $datum_registracije;
}

/**
 * @return mixed
 */
public function getDatumIzmene()
{
    return $this->datum_izmene;
}

/**
 * @param mixed $datum_izmene
 */
public function setDatumIzmene($datum_izmene): void
{
    $this->datum_izmene = $datum_izmene;
}

/**
 * @return mixed
 */
public function getObrisan()
{
    return $this->obrisan;
}

/**
 * @param mixed $obrisan
 */
public function setObrisan($obrisan): void
{
    $this->obrisan = $obrisan;
}

/**
 * @return mixed
 */
public function getIDGrad()
{
    return $this->ID_grad;
}

/**
 * @param mixed $ID_grad
 */
public function setIDGrad($ID_grad): void
{
    $this->ID_grad = $ID_grad;
}

/**
 * @return mixed
 */
public function getIDUloga()
{
    return $this->ID_uloga;
}
/**
 * @param mixed $ID_uloga
 */
public function setIDUloga($ID_uloga): void
{
    $this->ID_uloga = $ID_uloga;
}
public function login(){
    $query = DB::table($this->table)->where([["username",$this->username],
        ["password",md5($this->password)],
        ["aktivan",1],
        ["obrisan",0]
```



```
        ])->first();
        return $query;
    }
    public function register(){
        $query = DB::table($this->table)->insert([
            "ime" => $this->ime,
            "prezime" => $this->prezime,
            "email" =>$this->email,
            "username" => $this->username,
            "password" => md5($this->password),
            "token" => $this->token,
            "pol" => $this->pol,
            "datum_registracije" => $this->datum_registracije,
            "ID_grad" => $this->ID_grad,
            "ID_uloga" => 2
        ]);
        return $query;
    }
    public function getUsers(){
        $query = DB::table($this->table)->get();
        return $query;
    }
    public function insert(){
        $query = DB::table($this->table)->insert([
            "ime" => $this->ime,
            "prezime" => $this->prezime,
            "email" =>$this->email,
            "username" => $this->username,
            "password" => md5($this->password),
            "token" => $this->token,
            "pol" => $this->pol,
            "datum_registracije" => $this->datum_registracije,
            "ID_grad" => $this->ID_grad,
            "ID_uloga" => $this->ID_uloga,
            "aktivan" => $this->aktivan
        ]);
        return $query;
    }
    public function getOneUser(){
        $query = DB::table("korisnik as k")
            ->join("uloga as u","k.ID_uloga","=","u.ID_uloga")
            ->join("grad as g","k.ID_grad","=","g.ID_grad")
            ->where("ID_korisnik",$this->ID_korisnik)->get();
        return $query;
    }
    public function update(){
        $query = DB::table($this->table)->where("ID_korisnik",$this->ID_korisnik)
            ->update([
                "ime" => $this->ime,
                "prezime" => $this->prezime,
                "email" =>$this->email,
                "username" => $this->username,
                "password" => md5($this->password),
                "pol" => $this->pol,
                "datum_izmene" => $this->datum_izmene,
                "ID_grad" => $this->ID_grad,
                "ID_uloga" => $this->ID_uloga,
                "aktivan" => $this->aktivan,
                "obrisan" => $this->obrisan
            ]);
    }
    public function delete(){
        DB::table($this->table)->where("ID_korisnik", $this->ID_korisnik)-
>update(["obrisan" => 1]);
    }

    public function activate(){
        DB::table($this->table)->where("token", $this->token)->update([
           "aktivan" => 1
        ]);
    }

}
```

Link.php

```
<?php
/**
 * Created by PhpStorm.
 * User: ASUS
 * Date: 25.2.2019
 * Time: 22:44
 */

namespace App\Models;
use Illuminate\Support\Facades\DB;

class Link
{
```

```php
private $ID_link;
private $naziv;
private $putanja;
private $ID_link_parent;
private $table = 'link';

/**
 * @return mixed
 */
public function getIDLink()
{
    return $this->ID_link;
}

/**
 * @param mixed $ID_link
 */
public function setIDLink($ID_link): void
{
    $this->ID_link = $ID_link;
}

/**
 * @return mixed
 */
public function getNaziv()
{
    return $this->naziv;
}

/**
 * @param mixed $naziv
 */
public function setNaziv($naziv): void
{
    $this->naziv = $naziv;
}

/**
 * @return mixed
 */
public function getPutanja()
{
    return $this->putanja;
}
```

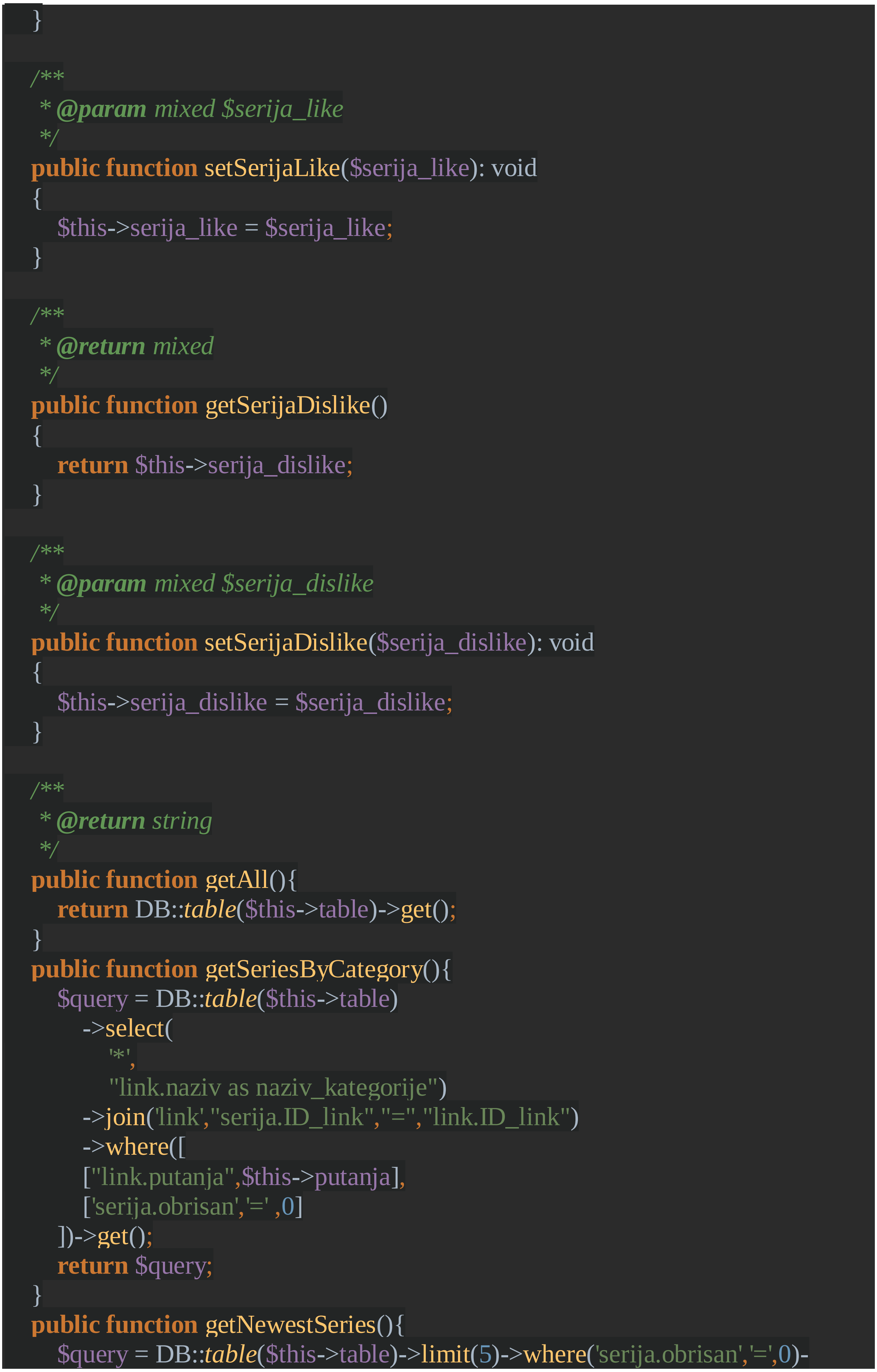
```
    /**
     * @param mixed $putanja
     */
    public function setPutanja($putanja): void
    {
        $this->putanja = $putanja;
    }

    /**
     * @return mixed
     */
    public function getIDLinkParent()
    {
        return $this->ID_link_parent;
    }

    /**
     * @param mixed $ID_link_parent
     */
    public function setIDLinkParent($ID_link_parent): void
    {
        $this->ID_link_parent = $ID_link_parent;
    }

    public function getMenu(){
        $query = DB::table($this->table)->where('ID_link_parent',-1)->get();
        return $query;
    }
    public function getCategories(){
        $query = DB::table($this->table)->where('ID_link_parent',0)-
>orderBy("naziv")->get();
        return $query;
    }
    public function insert(){
        DB::table($this->table)->insert([
                "naziv" => $this->naziv,
                "putanja" => $this->putanja,
                "ID_link_parent" => 0
            ]);
    }
    public function delete(){
        DB::table($this->table)->where("ID_link" , $this->ID_link)->delete();
    }
    public function getOne(){
        return DB::table($this->table)->where("ID_link", $this->ID_link)->get();
    }
    public function update(){
        return DB::table($this->table)->where("ID_link", $this->ID_link)->update([
            "naziv" => $this->naziv,
            "putanja" => $this->putanja
        ]);
    }
    public function getOtherCategoreis(){
        $query = DB::table($this->table)
            ->where('ID_link_parent',0)
            ->where('ID_link',"!=",$this->ID_link)
            ->orderBy("naziv")->get();
        return $query;
    }
}
```

Serija.php

```
<?php
/**
 * Created by PhpStorm.
 * User: Dušan
 * Date: 2.3.2019.
 * Time: 16.56
 */

namespace App\Models;

use Illuminate\Support\Facades\DB;

class Serija
{
    private $ID_serija;
    private $serija_naziv;
    private $serija_tekst;
    private $ID_serija_glumci;
    private $ID_link;
    private $godina;
    private $slika;
    private $obrisan;
    private $serija_like;
    private $serija_dislike;
    private $ID_korisnik;
    private $ID_glumci;
    private $data = [];
    /**
     * @param array $data
     */
    public function setData(array $data): void
    {
        $this->data = $data;
    }
    /**
     * @param mixed $ID_glumci
     */
    public function setIDGlumci($ID_glumci): void
    {
        $this->ID_glumci = $ID_glumci;
    }
    private $putanja;

    /**
     * @return mixed
     */
    public function getPutanja()
    {
        return $this->putanja;
    }

    /**
     * @param mixed $putanja
     */
    public function setPutanja($putanja): void
    {
        $this->putanja = $putanja;
    }


    /**
     * @return mixed
     */
    public function getIDKorisnik()
    {
        return $this->ID_korisnik;
    }

    /**
     * @param mixed $ID_korisnik
     */
    public function setIDKorisnik($ID_korisnik): void
    {
        $this->ID_korisnik = $ID_korisnik;
    }
    private $table = "serija";

    /**
     * @return mixed
     */
    public function getIDSerija()
    {
        return $this->ID_serija;
    }

    /**
     * @param mixed $ID_serija
     */
    public function setIDSerija($ID_serija): void
    {
        $this->ID_serija = $ID_serija;
    }

    /**
     * @return mixed
     */
    public function getSerijaNaziv()
    {
        return $this->serija_naziv;
    }

    /**
     * @param mixed $serija_naziv
     */
    public function setSerijaNaziv($serija_naziv): void
    {
        $this->serija_naziv = $serija_naziv;
    }

    /**
     * @return mixed
     */
    public function getSerijaTekst()
    {
        return $this->serija_tekst;
    }

    /**
     * @param mixed $serija_tekst
```

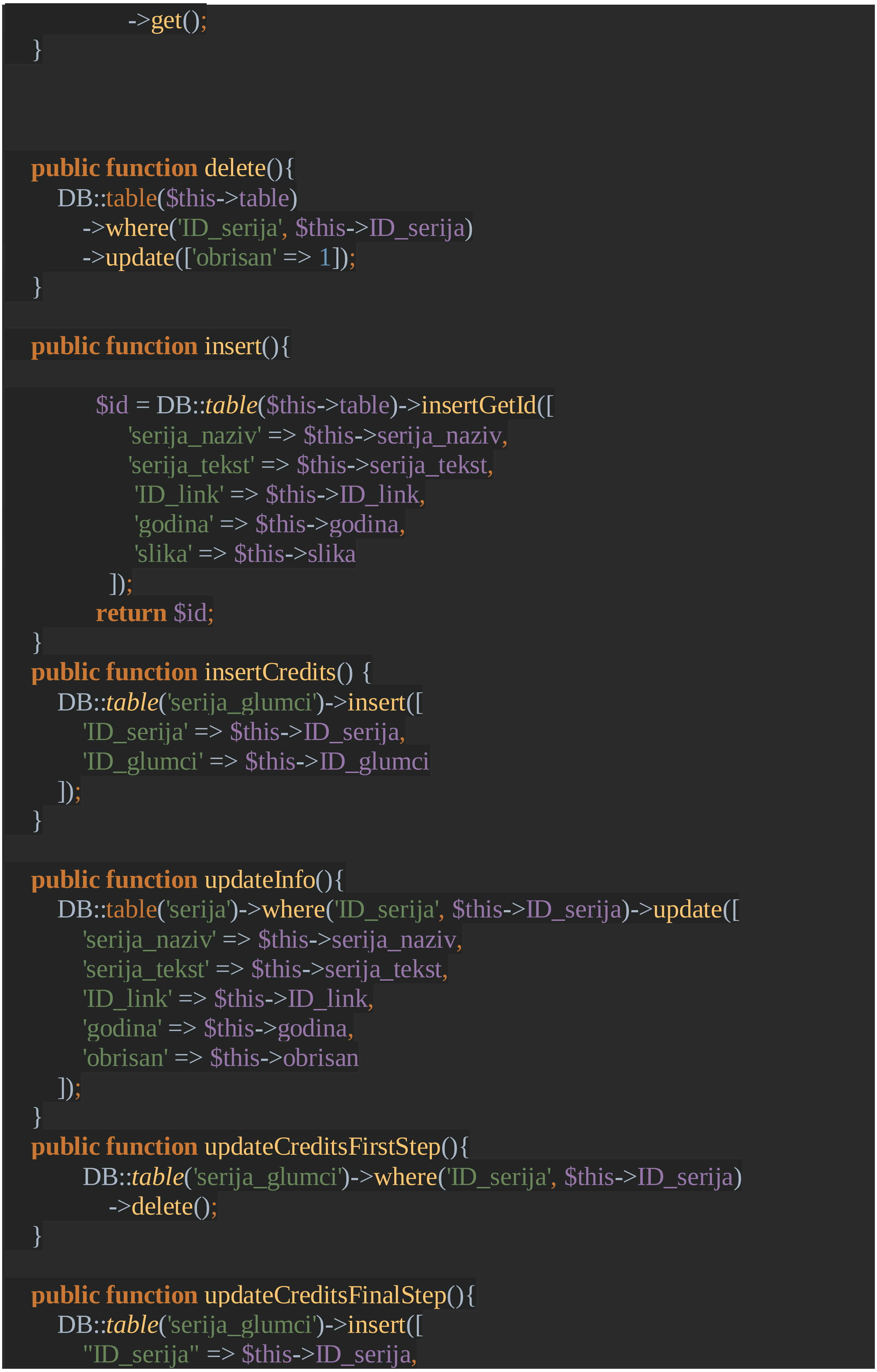
```php
 */
public function setSerijaTekst($serija_tekst): void
{
    $this->serija_tekst = $serija_tekst;
}

/**
 * @return mixed
 */
public function getIDSerijaGlumci()
{
    return $this->ID_serija_glumci;
}

/**
 * @param mixed $ID_serija_glumci
 */
public function setIDSerijaGlumci($ID_serija_glumci): void
{
    $this->ID_serija_glumci = $ID_serija_glumci;
}

/**
 * @return mixed
 */
public function getIDLink()
{
    return $this->ID_link;
}

/**
 * @param mixed $ID_link
 */
public function setIDLink($ID_link): void
{
    $this->ID_link = $ID_link;
}

/**
 * @return mixed
 */
public function getGodina()
{
    return $this->godina;
}
```

```
    /**
     * @param mixed $godina
     */
    public function setGodina($godina): void
    {
        $this->godina = $godina;
    }

    /**
     * @return mixed
     */
    public function getSlika()
    {
        return $this->slika;
    }

    /**
     * @param mixed $slika
     */
    public function setSlika($slika): void
    {
        $this->slika = $slika;
    }

    /**
     * @return mixed
     */
    public function getObrisan()
    {
        return $this->obrisan;
    }

    /**
     * @param mixed $obrisan
     */
    public function setObrisan($obrisan): void
    {
        $this->obrisan = $obrisan;
    }

    /**
     * @return mixed
     */
    public function getSerijaLike()
    {
        return $this->serija_like;
```



```
    }

    /**
     * @param mixed $serija_like
     */
    public function setSerijaLike($serija_like): void
    {
        $this->serija_like = $serija_like;
    }

    /**
     * @return mixed
     */
    public function getSerijaDislike()
    {
        return $this->serija_dislike;
    }

    /**
     * @param mixed $serija_dislike
     */
    public function setSerijaDislike($serija_dislike): void
    {
        $this->serija_dislike = $serija_dislike;
    }

    /**
     * @return string
     */
    public function getAll(){
        return DB::table($this->table)->get();
    }
    public function getSeriesByCategory(){
        $query = DB::table($this->table)
            ->select(
                '*',
                "link.naziv as naziv_kategorije")
            ->join('link',"serija.ID_link","=","link.ID_link")
            ->where([
            ["link.putanja",$this->putanja],
            ['serija.obrisan','=' ,0]
        ])->get();
        return $query;
    }
    public function getNewestSeries(){
        $query = DB::table($this->table)->limit(5)->where('serija.obrisan','=',0)-
```

```
>orderBy("godina", "desc")->get();
        return $query;
    }

    public function dislike()
    {
        DB::transaction(function () {
        DB::table("glasanje_serija")->insert([
            "ID_serija" => $this->ID_serija,
            "ID_korisnik" => $this->ID_korisnik
        ]);
        DB::table($this->table)
            ->where("ID_serija",$this->ID_serija)
            ->increment('serija_dislike' , 1);
        });
    }
    public function like()
    {
        DB::transaction(function () {
            DB::table("glasanje_serija")->insert([
                "ID_serija" => $this->ID_serija,
                "ID_korisnik" => $this->ID_korisnik
            ]);
            DB::table($this->table)
                ->where("ID_serija",$this->ID_serija)
                ->increment('serija_like' , 1);
        });
    }
    public function getOne(){
        return DB::table("serija as s")
            ->join("link as l",'s.ID_link',"=",'l.ID_link')
            ->where("s.ID_serija", $this->ID_serija)
            ->get();
    }
    public function getRoles(){
        return DB::table("glumci as g")
            ->join('serija_glumci as sg','g.ID_glumci',"=","sg.ID_glumci")
            ->join('serija as s',"sg.ID_serija","=","s.ID_serija")
            ->where("s.ID_serija", $this->ID_serija)
            ->get();
    }

    public function search(){
        return DB::table($this->table)
                   ->where("serija_naziv","LIKE", "%$this->serija_naziv%")
                   ->where('obrisan',"=",0)
```



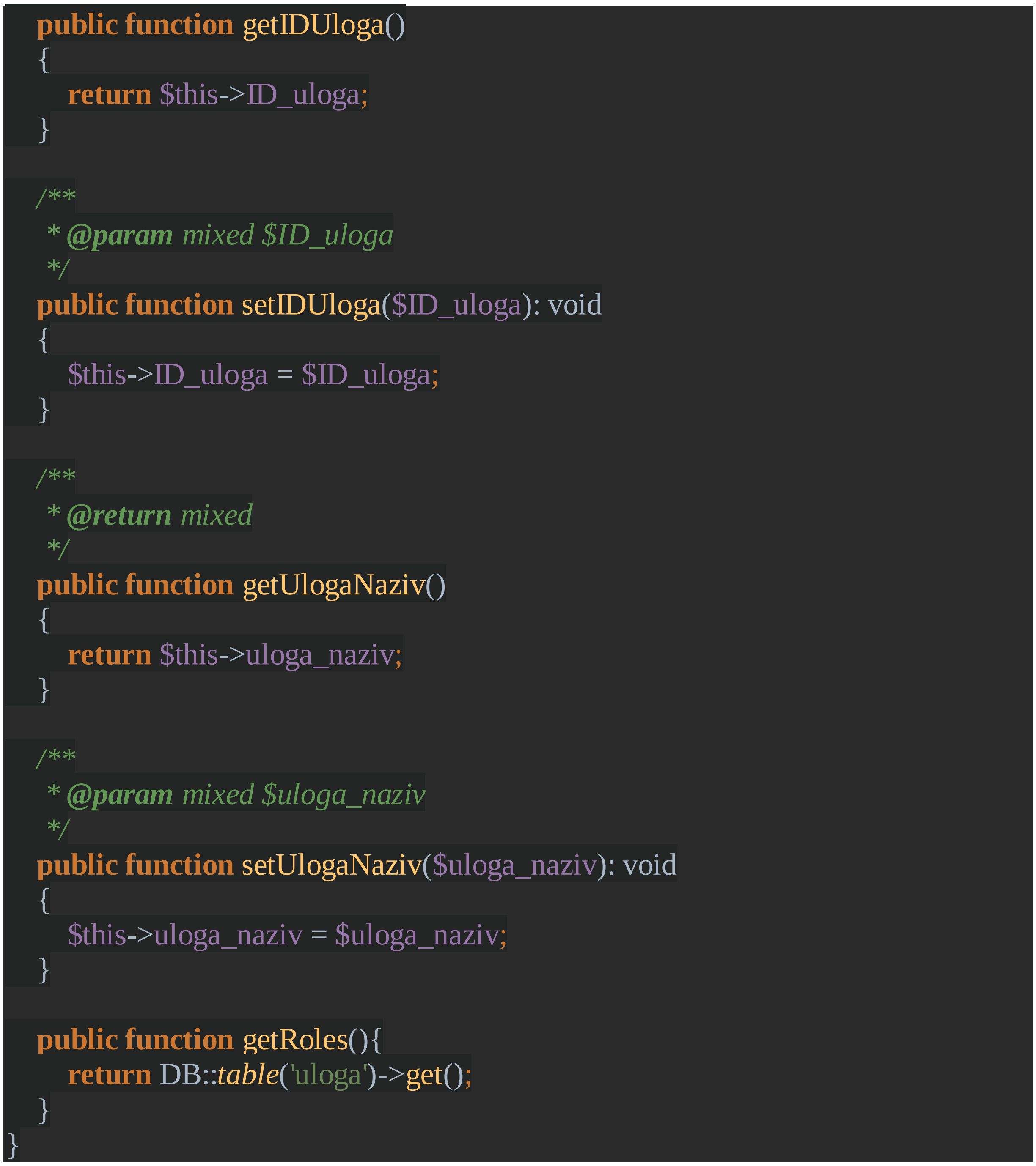
```
                ->get();
    }



    public function delete(){
        DB::table($this->table)
            ->where('ID_serija', $this->ID_serija)
            ->update(['obrisan' => 1]);
    }

    public function insert(){

             $id = DB::table($this->table)->insertGetId([
                  'serija_naziv' => $this->serija_naziv,
                  'serija_tekst' => $this->serija_tekst,
                   'ID_link' => $this->ID_link,
                   'godina' => $this->godina,
                   'slika' => $this->slika
               ]);
             return $id;
    }
    public function insertCredits() {
        DB::table('serija_glumci')->insert([
            'ID_serija' => $this->ID_serija,
            'ID_glumci' => $this->ID_glumci
        ]);
    }

    public function updateInfo(){
        DB::table('serija')->where('ID_serija', $this->ID_serija)->update([
            'serija_naziv' => $this->serija_naziv,
            'serija_tekst' => $this->serija_tekst,
            'ID_link' => $this->ID_link,
            'godina' => $this->godina,
            'obrisan' => $this->obrisan
        ]);
    }
    public function updateCreditsFirstStep(){
            DB::table('serija_glumci')->where('ID_serija', $this->ID_serija)
                ->delete();
    }

    public function updateCreditsFinalStep(){
        DB::table('serija_glumci')->insert([
            "ID_serija" => $this->ID_serija,
            'ID_glumci' => $this->ID_glumci
        ]);
    }


    public function updatePicture(){
        DB::table('serija')->where('ID_serija', $this->ID_serija)->update([
           'slika' => $this->slika
        ]);
    }

}
```

Sponzor.php

```
<?php
/**
 * Created by PhpStorm.
 * User: Dušan
 * Date: 2.3.2019.
 * Time: 14.07
 */

namespace App\Models;

use Illuminate\Support\Facades\DB;

class Sponzor
{
    private $ID_sponsor;
    private $slika;
    private $link;

    /**
     * @return mixed
     */
    public function getLink()
    {
        return $this->link;
    }

    /**
     * @param mixed $link
     */
    public function setLink($link): void
    {
        $this->link = $link;
    }
    private $table = "sponzor";

    /**
     * @return mixed
     */
    public function getIDSponsor()
    {
        return $this->ID_sponsor;
    }

    /**
     * @param mixed $ID_sponsor
     */
    public function setIDSponsor($ID_sponsor): void
    {
        $this->ID_sponsor = $ID_sponsor;
    }

    /**
     * @return string
     */

    /**
     * @return mixed
     */
    public function getSlika()
    {
        return $this->slika;
    }

    /**
     * @param mixed $slika
     */
    public function setSlika($slika): void
    {
        $this->slika = $slika;
    }
    public function getSponsors(){
        $query = DB::table($this->table)->get();
        return $query;
    }
    public function insertSponsor(){
        DB::table($this->table)->insert([
            "slika" => $this->slika,
            "link" => $this->link
        ]);
    }
    public function deleteSponsor(){
        DB::table($this->table)->where("ID_sponzor",$this->ID_sponsor)->delete();
    }
    public function getOneSponsor(){
        $query =  DB::table($this->table)->where('ID_sponzor',$this->ID_sponsor)-
>get();
        return $query;
    }
    public function updateLink(){
         DB::table("sponzor")->where("ID_sponzor",$this->ID_sponsor)-
>update(["link" => $this->link]);
    }
    public function updatePicture(){
        DB::table("sponzor")->where("ID_sponzor",$this->ID_sponsor)-
>update(["slika" => $this->slika]);
    }
}
```

Uloga.php

```
<?php
/**
 * Created by PhpStorm.
 * User: Dušan
 * Date: 7.3.2019.
 * Time: 13.47
 */

namespace App\Models;


use Illuminate\Support\Facades\DB;

class Uloga
{
    private $ID_uloga;
    private $uloga_naziv;

    /**
     * @return mixed
     */
```



```
    public function getIDUloga()
    {
        return $this->ID_uloga;
    }

    /**
     * @param mixed $ID_uloga
     */
    public function setIDUloga($ID_uloga): void
    {
        $this->ID_uloga = $ID_uloga;
    }

    /**
     * @return mixed
     */
    public function getUlogaNaziv()
    {
        return $this->uloga_naziv;
    }

    /**
     * @param mixed $uloga_naziv
     */
    public function setUlogaNaziv($uloga_naziv): void
    {
        $this->uloga_naziv = $uloga_naziv;
    }

    public function getRoles(){
        return DB::table('uloga')->get();
    }
}
```

----------------------------------public/css--

main.css

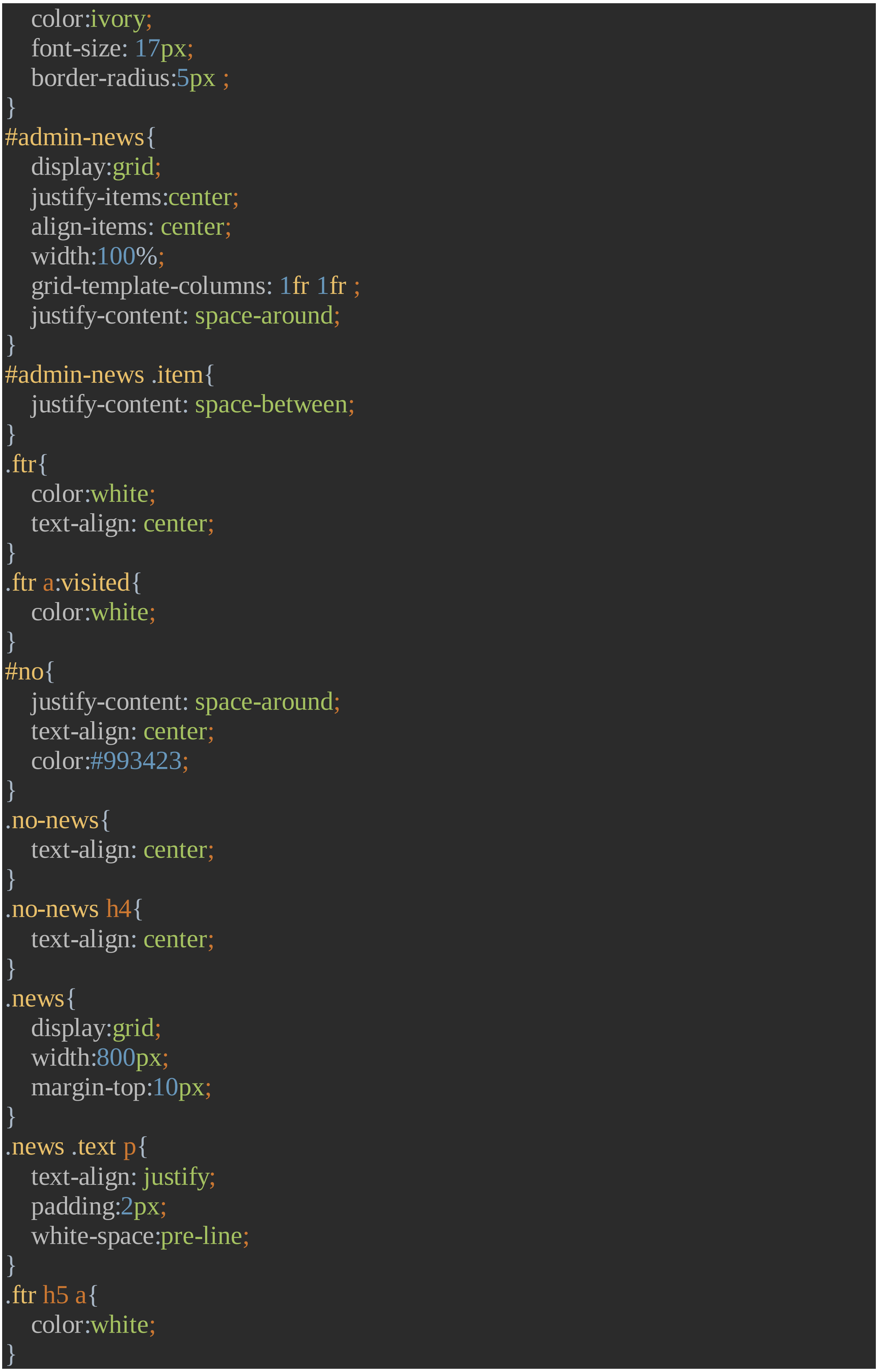
```
*{
   margin:0;
   padding:0;
   font-family: "Segoe UI Semibold";
}
a{
   text-decoration: none;
}
img{
   border-radius: 3px;
}
#wrapper{
   display:flex;
   flex-direction: column;
   width:100%;
}
#header{
   display:flex;
   width:100%;
   border-bottom: 4px #1b4b72 solid;
   justify-content: space-around;
}
#logo{
   flex-basis:180px;
   height:98px;
}
li{
   list-style: none;
}
nav{
   display:flex;
   justify-items: center;
   align-items: center;
   flex-basis:55%;
   margin-bottom:1px;
   max-height: 98px;
   margin-left: 60px;
}
nav > ul{
   display:flex;
   flex-basis:100%;
   justify-content: space-between;
```

```
}
nav > ul > li{
    display:flex;
    padding:0 15px;
    background: #1b4b72;
    justify-content: space-between;
    align-content: space-between;
    max-height: 98px;
    border-radius: 0 0 10px 10px;
}
nav > ul > li > a{
    display:flex;
    text-align: center;
    padding:35px 0;
    color:ivory;
    font-size:20px;
    justify-self: center;
    align-self: center;
    font-family: "Segoe UI Semibold";
}
nav > ul > li > h4{
    display:flex;
    text-align: center;
    padding:35px 0;
    color:ivory;
    font-size:20px;
    justify-self: center;
    align-self: center;
    font-family: "Segoe UI Semibold";
}
nav ul li:hover{
    animation: nav ;
    animation-duration: 1s;
}
@keyframes nav{
    from{background-color: #1b4b72; }
    to{ background-color: #1b77aa; }
}
#criteria {
    flex-basis: 20%;
    display: flex;
    align-items: center;
    margin-left: 30px;
}
#search{
    margin:20px;
    padding:4px;
    font-size: 18px;
    outline: #00adff;
    border-radius: 5px;
}
.fa-search{
    justify-content: flex-end;
}
#container{
    display: grid;
    grid-template-areas: "left-side articles right-side";
    width:100%;
    justify-content: space-between;
}
#left-side{
    display:flex;
    width:20%;
}
#left-side ul {
    display:flex;
    flex-direction: column;
    justify-items: center;
    align-content: space-around;
    align-items: center;
}
#left-side ul li{
    padding:20px;
    border-radius:0px 20px 20px 0px;
    justify-self: center;
    justify-content: space-between;
    margin-top:30px;
    background-color: #b75800;
    width:150px;

}
#left-side ul li:first-child{
    margin-top:1px;
}
#left-side > ul > li > a> h3{
    color:ivory;
    font-size:28px;
    cursor: pointer;
    font-weight: normal;
}
#articles{
    display:grid;
```

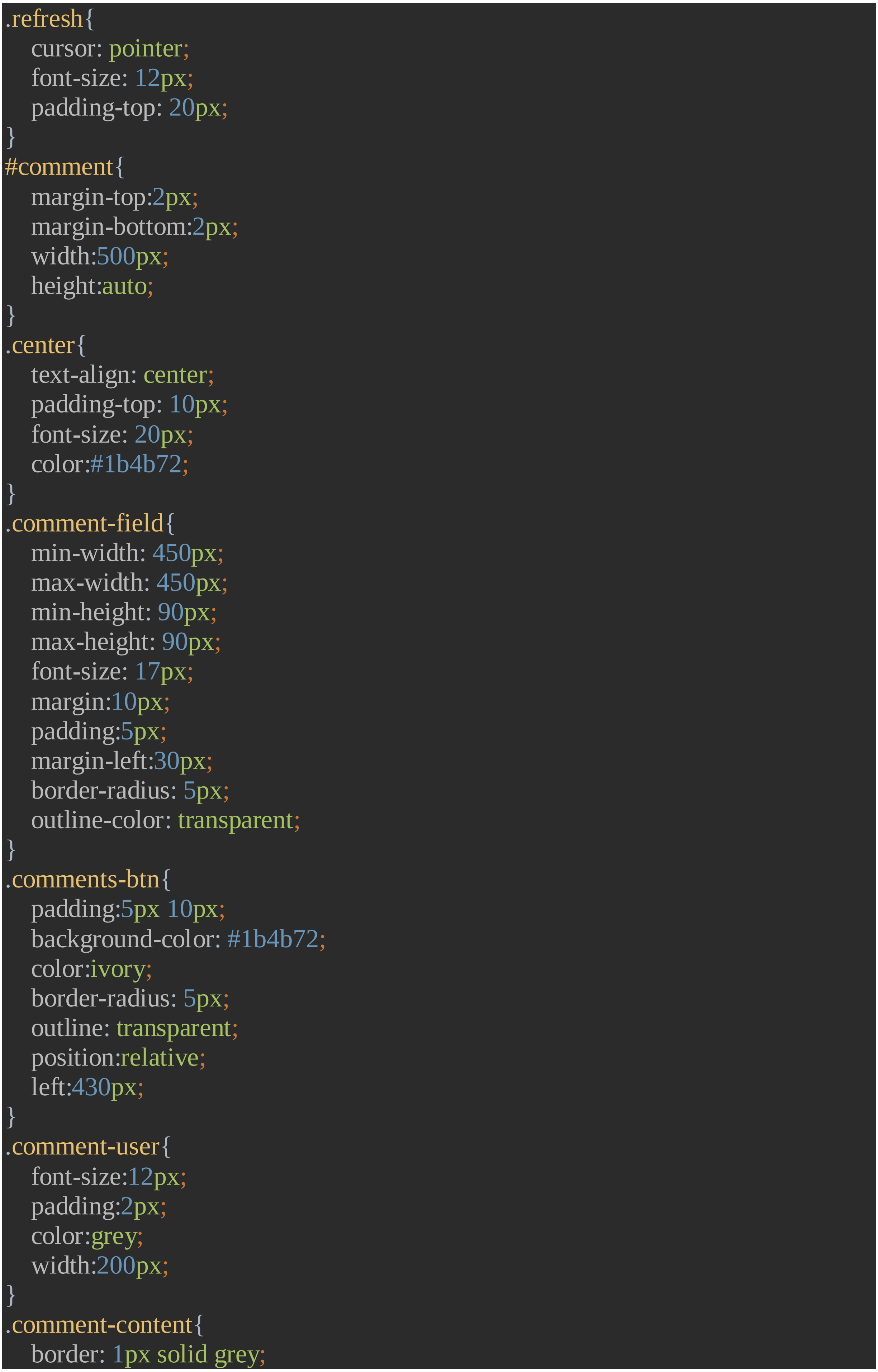
```
   width:95%;
   align-content: baseline;
   justify-content: space-around;
}
.item{
   display:grid;
   flex-direction: row;
   grid-template-columns: 1fr 2fr;
   grid-template-areas: "img right-container";
   max-height: max-content;
   justify-content: space-around;
   align-items: baseline;
   width:100%;
   margin-top:20px;
}

.img{
   align-self: center;
   justify-self: right;
   margin:0 30px;
}
.heading{
   margin-top:30px;
   text-align:center;
}
.text{
   margin-top: 10px;
   word-break:keep-all;
   text-align:justify;
}
.date-com{
   display:flex;
   flex-basis:100%;
}
.fa-plus{
   color:lightgreen;
   cursor:pointer;
}
.fa-minus{
   cursor:pointer;
   color:lightcoral;
   margin-left:10px;
}
.right{
   cursor: pointer;
}
```

```
.details{
    cursor: pointer;
    word-break: break-all;
}
.details:hover{
    text-decoration: underline;
}
.no-underline:hover{
    text-decoration: none;
    cursor: auto;
}
#right-side{
    display:flex;
    width:100%;
    flex-direction: column;
}
.sponsor{
    margin:2px 5px;
    flex-basis: 100%;
    max-height: max-content;
}
footer{
    display:flex;
    background-color: #1b4b72;
    width:100%;
    height:100%;
    justify-content: center;
    margin-top:2px;
    justify-items: center;
    align-items: center;
}
footer h6{
    justify-self: center;
    color:ivory;
    padding:5px;
    font-size:14px;
}
footer a{
    color:ivory;
    justify-self: flex-start;
    padding:20px 10px;
    font-size: 15px;
}
#autor{
    display:flex;
    width:100%;
    justify-content: center;
    align-content: center;
    margin-top:160px;
}
#autor_slika img{
    border-radius: 20px;
    justify-self: center;
    align-self: center;
}
#autor_tekst{
    justify-self: center;
    align-self: center;
    width:50%;
    padding:10px;
    font-size: 17px;
}
#autor_tekst h4{
    text-align: justify;
    color:#1b4b72;
    padding:10px;
    font-size:20px;
    background-color:#e8edef;
    border-radius: 20px;
}
#kontakt input[type="text"]{
    margin:10px;
}
#kontakt_forma{
    display:flex;
    width:600px;
    text-align:center;
    margin:0 auto;
    padding:20px;
}
#kontakt_forma input {
    padding:5px;
    border-radius:5px;
    width:200px;
    margin-top: 20px;
}
#ime_korisnik{
    width:60%;
}
#prezime_korisnik{
    width:60%;
}
```

```
#kontakt_forma textarea{
    margin-top:20px;
    width:400px;
    height:250px;
    min-height: 250px;
    max-height: 250px;
    min-width: 400px;
    max-width: 400px;
    border-radius:10px;
    padding:10px;
    font-size:18px;
    margin-bottom: 5px;
}
.btn {
    padding: 10px 100px 10px 100px;
    font-size: 20px;
    background-color: #333333;
    color: #e5e5e5;
    border-radius: 20px;
    outline-color: transparent;
    margin: 0 auto;
}
.btn:hover{
    animation-name: btn;
    animation-duration: 1s;
    background: #1b4b72;
}
#kontaktOk{
    display:none;
}
.edit{
    padding:2px;
    background-color: #1b4b72;
    text-align: center;
    color:ivory;
    display:inline-block;
    border-radius:5px ;
    font-size: 17px;
    outline: transparent;
    position: sticky;
}
.del{
    padding:2px;
    background-color: darkred;
    text-align: center;
    margin-top:3px;
```



```
   color:ivory;
   font-size: 17px;
   border-radius:5px ;
}
#admin-news{
   display:grid;
   justify-items:center;
   align-items: center;
   width:100%;
   grid-template-columns: 1fr 1fr ;
   justify-content: space-around;
}
#admin-news .item{
   justify-content: space-between;
}
.ftr{
   color:white;
   text-align: center;
}
.ftr a:visited{
   color:white;
}
#no{
   justify-content: space-around;
   text-align: center;
   color:#993423;
}
.no-news{
   text-align: center;
}
.no-news h4{
   text-align: center;
}
.news{
   display:grid;
   width:800px;
   margin-top:10px;
}
.news .text p{
   text-align: justify;
   padding:2px;
   white-space:pre-line;
}
.ftr h5 a{
   color:white;
}
```



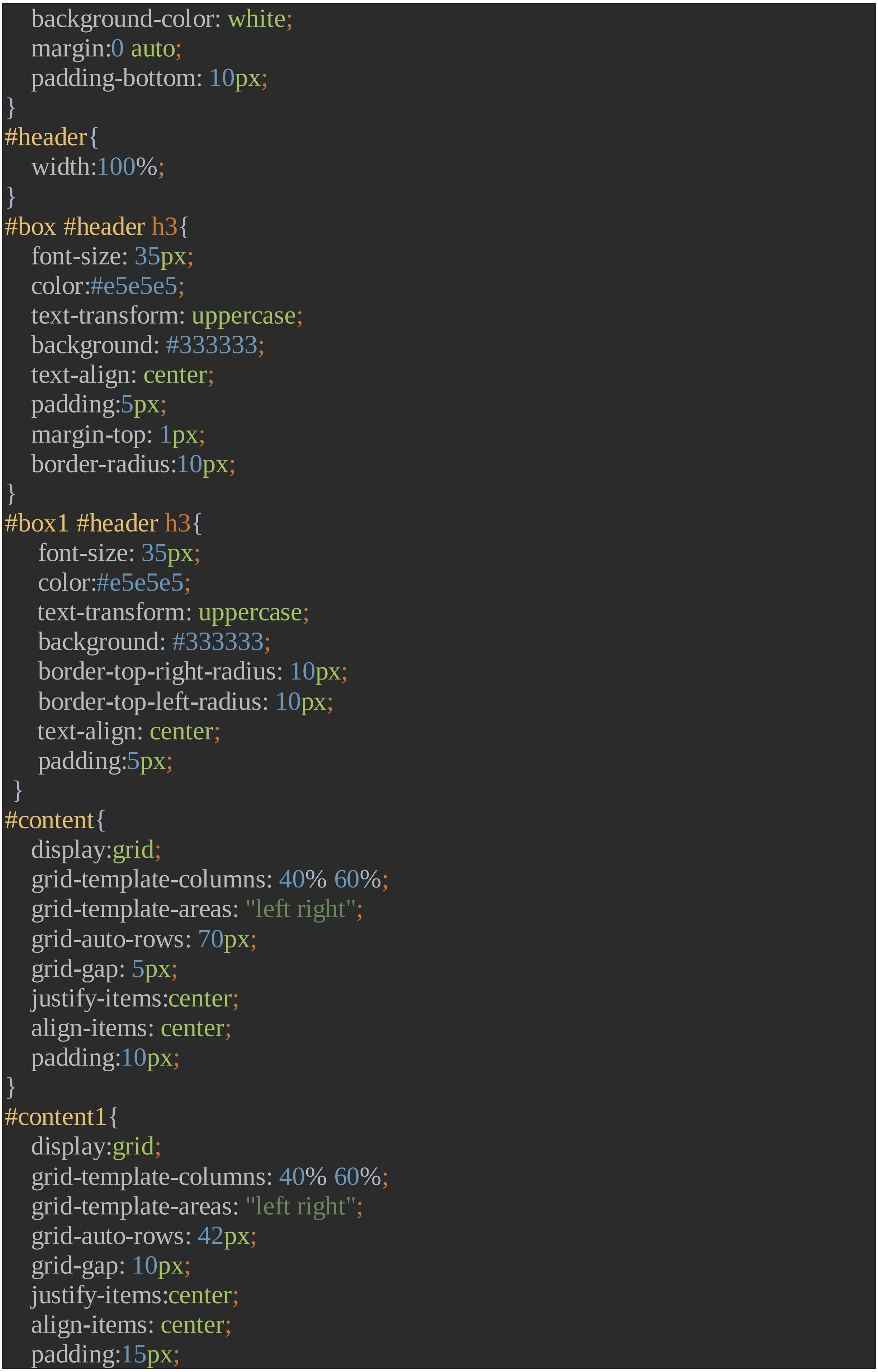
```
.refresh{
   cursor: pointer;
   font-size: 12px;
   padding-top: 20px;
}
#comment{
   margin-top:2px;
   margin-bottom:2px;
   width:500px;
   height:auto;
}
.center{
   text-align: center;
   padding-top: 10px;
   font-size: 20px;
   color:#1b4b72;
}
.comment-field{
   min-width: 450px;
   max-width: 450px;
   min-height: 90px;
   max-height: 90px;
   font-size: 17px;
   margin:10px;
   padding:5px;
   margin-left:30px;
   border-radius: 5px;
   outline-color: transparent;
}
.comments-btn{
   padding:5px 10px;
   background-color: #1b4b72;
   color:ivory;
   border-radius: 5px;
   outline: transparent;
   position:relative;
   left:430px;
}
.comment-user{
   font-size:12px;
   padding:2px;
   color:grey;
   width:200px;
}
.comment-content{
   border: 1px solid grey;
```

```
    border-radius: 5px;
    padding:4px 10px;
    width:400px;
    word-break: break-all;
}
.warning{
    color:ivory;
    background-color: #993423;
    margin-top:5px;
    text-align: center;
    padding:5px;
    display: none;
    border-radius: 20px;
}
.success{
    border-radius: 20px;
    color:ivory;
    margin-top:5px;
    text-align: center;
    padding:5px;
    display: none;
    background-color: darkgreen;
}
.forbidden{
    padding:5px 0;
}
.forbidden a{
    text-align: center;
    color:#ac1412;
}
.anketa{
    width:500px;
    padding-top:3px;
}
.fa-thumbs-down{
    color:#993423;
    cursor: pointer;
    margin-left:5px;
    width:max-content;
}
.fa-thumbs-up{
    margin-left:5px;
    color:green;
    cursor: pointer;
    width:max-content;
}
```

```
.fa-thumbs-down:hover{
    color: #dc141f;
}
.fa-thumbs-up:hover{
    color: limegreen;
}
.one{
    width:400px;
}
.red{
    color:red;
    margin-left:10px;
}
.green{
    color:green;
}
.fa-trash-alt{
    width:100px;
    text-align: right;
    padding-top:3px;
    cursor: pointer;
}
.comment-heading{
    font-size: 17px;
    color:#9e9e9e;
    width:300px;
    text-align: center;
}
.feedbackAdmin{
    text-align: center;
    color:#e5e5e5;
    padding:5px;
    font-size: 17px;
    font-weight: lighter;
    width:100%;
}
.ok{
    background:  linear-gradient(45deg, #00adff,#0077ff);
}
.er{
    background: linear-gradient(45deg, crimson ,#ac1412);
}
.date-com .fa{
    justify-items: left;
}
.left{
   width:max-content;
}
.right{
   position:relative;
   left:400px;
   font-size:12px;
}
.roles{
   display:inline-block;
}
.big{
   padding-bottom:20px;
}
```

main-login-reg.css

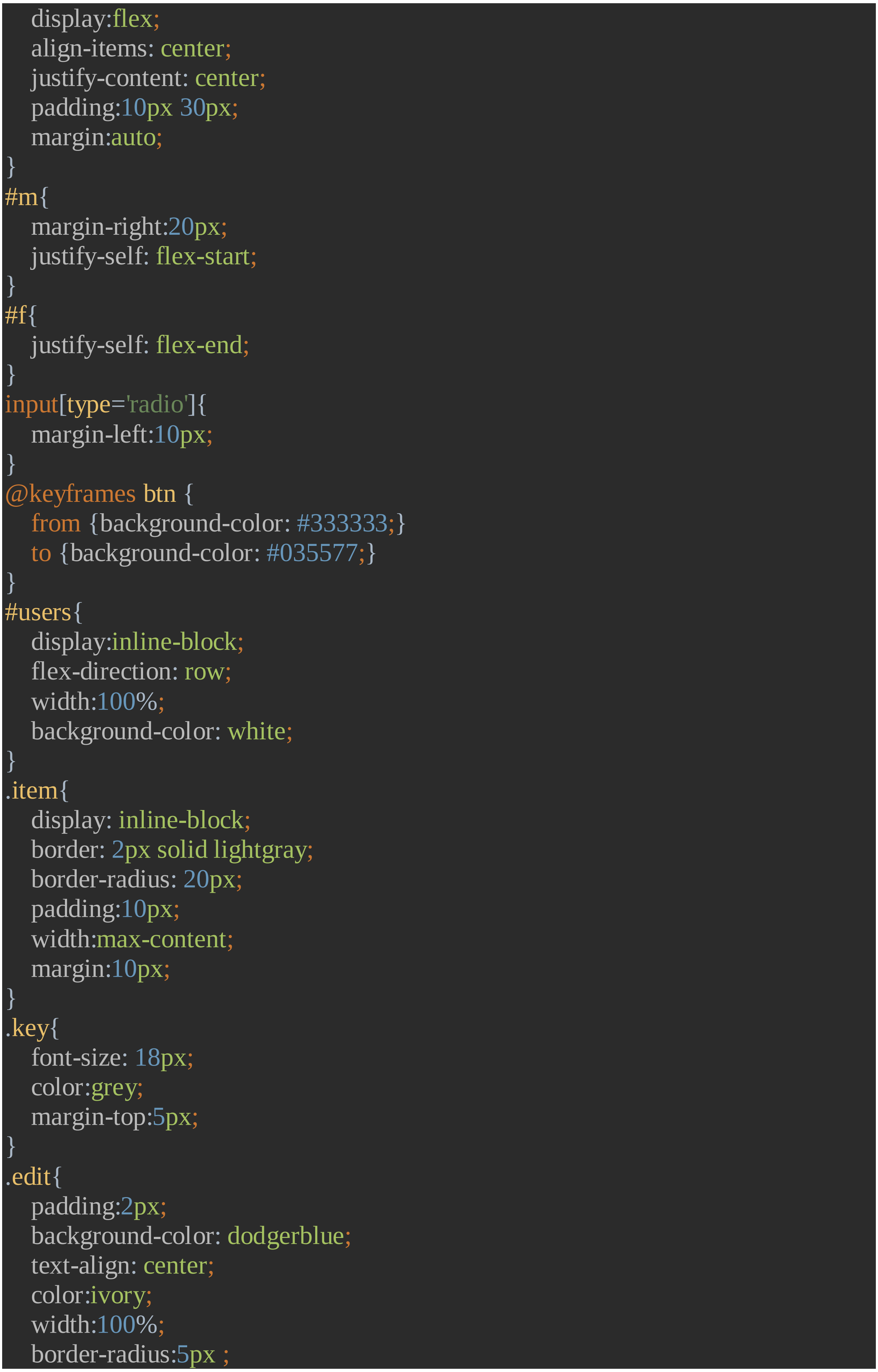
```
*{
   padding: 0;
   margin: 0;
   font-family: "Segoe UI Semibold";
}
#wrapper{
   display: flex;
   flex-basis: 100%;
   min-height: 100vh;
   background:   -webkit-linear-gradient(45deg, #0091a0bf,#1b4b72);
   background: -o-linear-gradient(45deg, #0091a0bf,#1b4b72);
   background: -moz-linear-gradient(45deg,#0091a0bf,#1b4b72);
   align-items: center;
   justify-content: center;
   justify-self: center;
}
#box{
   display: block;
   width:412px;
   border:2px white solid;
   border-radius: 10px;
   margin:0 auto;
   background: white;
   padding-bottom: 10px;
}
#box1{
   display: block;
   width:410px;
   min-height:100%;
```



```
    background-color: white;
    margin:0 auto;
    padding-bottom: 10px;
}
#header{
    width:100%;
}
#box #header h3{
    font-size: 35px;
    color:#e5e5e5;
    text-transform: uppercase;
    background: #333333;
    text-align: center;
    padding:5px;
    margin-top: 1px;
    border-radius:10px;
}
#box1 #header h3{
     font-size: 35px;
     color:#e5e5e5;
     text-transform: uppercase;
     background: #333333;
     border-top-right-radius: 10px;
     border-top-left-radius: 10px;
     text-align: center;
     padding:5px;
 }
#content{
    display:grid;
    grid-template-columns: 40% 60%;
    grid-template-areas: "left right";
    grid-auto-rows: 70px;
    grid-gap: 5px;
    justify-items:center;
    align-items: center;
    padding:10px;
}
#content1{
    display:grid;
    grid-template-columns: 40% 60%;
    grid-template-areas: "left right";
    grid-auto-rows: 42px;
    grid-gap: 10px;
    justify-items:center;
    align-items: center;
    padding:15px;
```

```
}

.right{
   padding:10px;
   font-size:15px;
   align-self: center;
   justify-self: end;
   width:83%;
   border-radius:5px;
   background:white;
   border: 1px groove #9e9e9e;
   outline-color:transparent;
}
.left{
   align-self:center;
   justify-self: start;
   font-size: 17px;
   font-weight: bolder;
   text-transform: initial;
}
.btn{
   padding:10px 100px 10px 100px ;
   font-size:20px;
   background-color:  #333333;
   color:#e5e5e5;
   border-radius: 20px;
   outline-color: transparent;
   margin-left:238px;
}
.btn:hover{
   animation-name: btn;
   animation-duration: 1s;
   background: #035577;
}
.btn:active{
   position:relative;
   bottom:-3px;
   left:3px;
}
h5{
   text-align: center;
   font-size:15px;
   color:#999;
}
h5 a{
   color:#999;
```

```
}
a{
    text-decoration: none;
    color: #333333;
}
a:visited{
    color:#333333;
}
.select{
    width: 92.7%;
}
.feedback{
    text-align: center;
    color:#e5e5e5;
    padding:5px;
    border-top-right-radius:10px;
    border-top-left-radius:10px;
    font-size: 17px;
    font-weight: lighter;
    display: none;
}
.feedbackAdmin{
    text-align: center;
    color:#e5e5e5;
    padding:5px;
    border-top-right-radius:10px;
    border-top-left-radius:10px;
    font-size: 17px;
    font-weight: lighter;
}
.ok{
    background:  linear-gradient(45deg, #00adff,#0077ff);
}
.er{
    background: linear-gradient(45deg, crimson ,#ac1412);
}
.false{
    padding:5px;
    text-align: center;
    color:#e5e5e5;
    border-top-right-radius:10px;
    border-top-left-radius:10px;
    font-size: 14.5px;
    margin-bottom:1px;
}
#pol{
```



```
    display:flex;
    align-items: center;
    justify-content: center;
    padding:10px 30px;
    margin:auto;
}
#m{
    margin-right:20px;
    justify-self: flex-start;
}
#f{
    justify-self: flex-end;
}
input[type='radio']{
    margin-left:10px;
}
@keyframes btn {
    from {background-color: #333333;}
    to {background-color: #035577;}
}
#users{
    display:inline-block;
    flex-direction: row;
    width:100%;
    background-color: white;
}
.item{
    display: inline-block;
    border: 2px solid lightgray;
    border-radius: 20px;
    padding:10px;
    width:max-content;
    margin:10px;
}
.key{
    font-size: 18px;
    color:grey;
    margin-top:5px;
}
.edit{
    padding:2px;
    background-color: dodgerblue;
    text-align: center;
    color:ivory;
    width:100%;
    border-radius:5px ;
    font-size: 17px;
    outline: transparent;
    position: sticky;
}
.del{
    padding:2px;
    background-color: darkred;
    text-align: center;
    margin-top:3px;
    color:ivory;
    width:100%;
    font-size: 17px;
    border-radius:5px ;
}
.sp{
    width:300px;
}
#edit-news{
    padding:5px 5px;
}
#edit-news textarea{
    min-width:400px;
    max-width: fit-content;
    min-height:200px;
    max-height: 200px;
}
.btn1{
    position: relative;
    left:-190px;
    margin-top:20px;
}
.btn1:active{
    position: relative;
    left:-190px;
    margin-top:20px;
}
.btn2{
    margin-left:70px;
    margin-top:10px;
}
#sponsors{
    display:flex;
    flex-direction: row;
    flex-wrap: wrap;
}
.sponsorAdmin{
    width:300px;
    margin:2px 10px;
    padding:5px 10px;
    border: 1px solid #9e9e9e;
    border-radius: 2px;
}
#sponsors button{
    padding:5px;
    background-color: darkred;
    color:ivory;
    margin-bottom:10px;
    outline: transparent;
    border-radius: 10px;
}
.categoryAdmin{
    width:max-content;
    margin:2px 10px;
    padding:5px 10px;
    border: 1px solid #9e9e9e;
    border-radius: 2px;
    text-align: center;
}
#sponsors .editSponsor{
    background-color: #1b77aa;
}
.categoryAdmin h3{
    text-align: center;
}
.series{
    display:flex;
    width:max-content;
    padding-top:10px;
}
.padd{
    padding-left:20px;
}
.s{
    width:80px;
    margin-top:15px;
}
.row{
    padding:10px;
    text-align: center;
}
.cell{
    padding:10px;
```

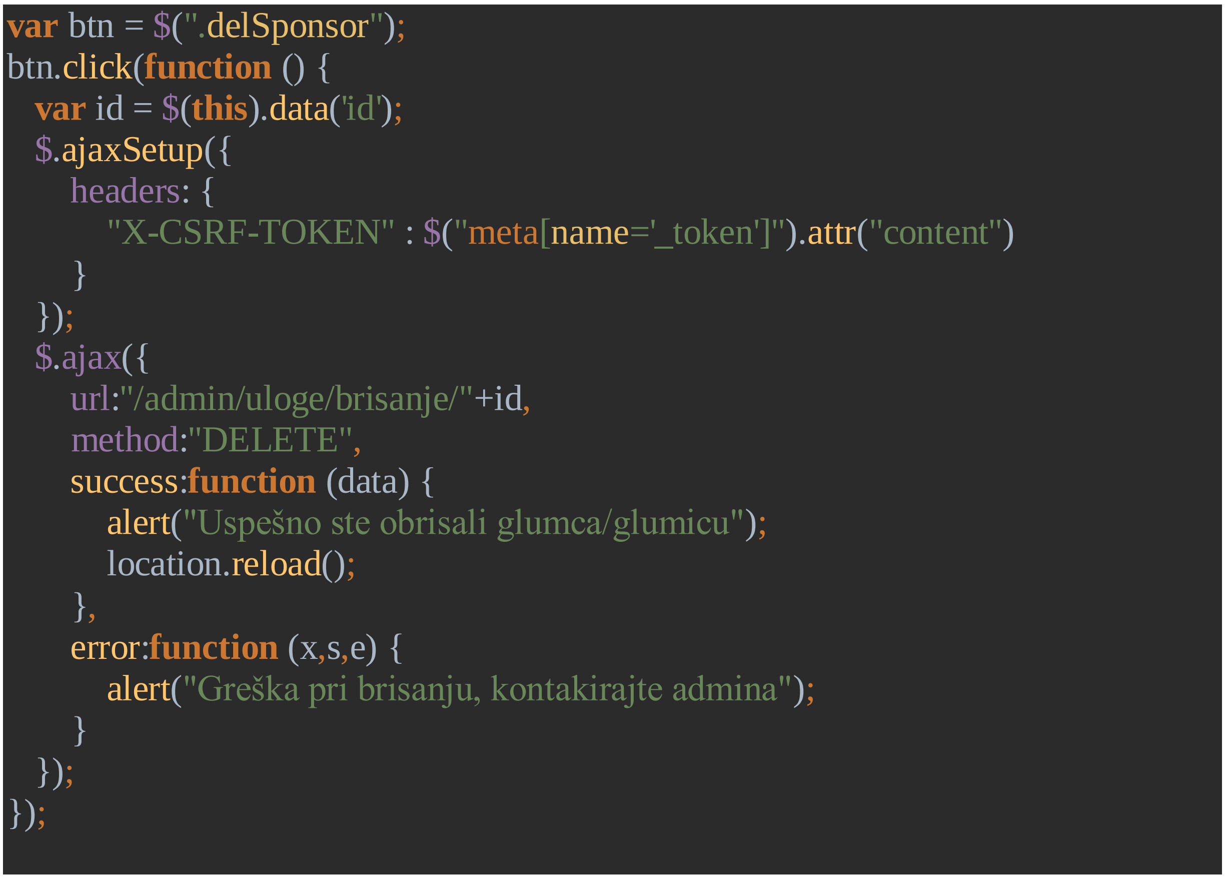
```
   text-align: center;
}
.root{
   font-weight: bolder;
}
```

-----------------------------------public/js--

adminDeleteCategory.js

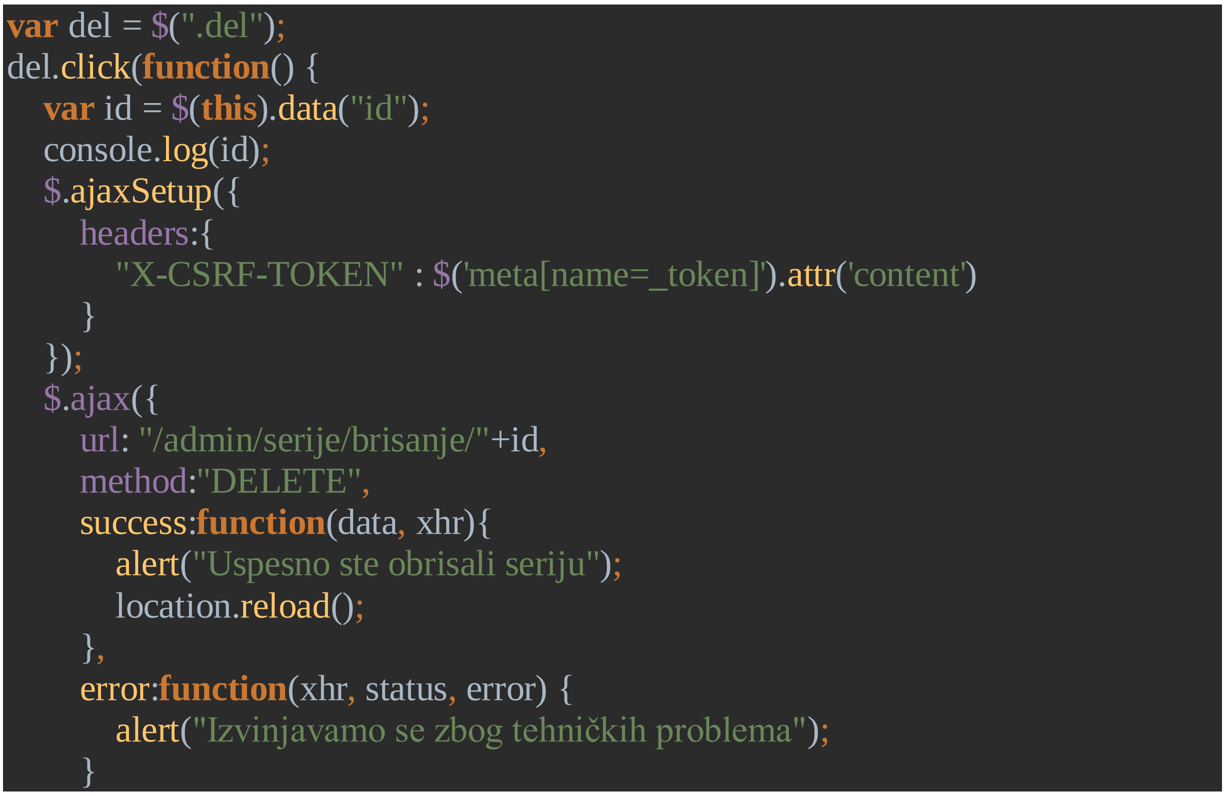
```
var del = $(".delSponsor");
del.click(function() {
   var id = $(this).data("id");
   console.log(id);
   $.ajaxSetup({
      headers:{
         "X-CSRF-TOKEN" : $('meta[name=_token]').attr('content')
      }
   });
   $.ajax({
      url: "/admin/zanrovi/brisanje/"+id,
      method:"DELETE",
      success:function(data, xhr){
         alert("Uspesno ste obrisali kategoriju");
         location.reload();
      },
      error:function(xhr, status, error) {
         switch (xhr.status) {
            case 500:
               alert("Izvinjavamo se zbog tehničkih problema");
               break;
         }
      }
   });
});
```

adminDeleteRole.js



```
var btn = $(".delSponsor");
btn.click(function () {
  var id = $(this).data('id');
  $.ajaxSetup({
    headers: {
      "X-CSRF-TOKEN" : $("meta[name='_token']").attr("content")
    }
  });
  $.ajax({
    url:"/admin/uloge/brisanje/"+id,
    method:"DELETE",
    success:function (data) {
      alert("Uspešno ste obrisali glumca/glumicu");
      location.reload();
    },
    error:function (x,s,e) {
      alert("Greška pri brisanju, kontakirajte admina");
    }
  });
});
```

adminDeleteSeries.js



```
var del = $(".del");
del.click(function() {
  var id = $(this).data("id");
  console.log(id);
  $.ajaxSetup({
    headers:{
      "X-CSRF-TOKEN" : $('meta[name=_token]').attr('content')
    }
  });
  $.ajax({
    url: "/admin/serije/brisanje/"+id,
    method:"DELETE",
    success:function(data, xhr){
      alert("Uspesno ste obrisali seriju");
      location.reload();
    },
    error:function(xhr, status, error) {
      alert("Izvinjavamo se zbog tehničkih problema");
    }
    });
});
```

adminDeleteUser.js

```
var del = $(".del");
del.click(function() {
    var id = $(this).data("id");
    console.log(id);
    $.ajaxSetup({
        headers:{
            "X-CSRF-TOKEN" : $('meta[name=_token]').attr('content')
        }
    });
    $.ajax({
        url: "/admin/korisnici/brisanje/"+id,
        method:"DELETE",
        success:function(data, xhr){
            alert("Uspesno ste obrisali korisnika");
            location.reload();
        },
        error:function(xhr, status, error) {
            switch (xhr.status) {
                case 500:
                    alert("Izvinjavamo se zbog tehničkih problema");
                    break;
            }
        }
    });
});
```

adminInsertCategory.js

```
var btn = $("#btn");
var category = $("#category");
var feeedbackOk = $("#f200");
var feedbackError1 = $("#f500");
var div = $("#errors");
var er = $(".er");
var falsee = $(".false");
var errors = [];
btn.click(() => {
    var reCategory = /^[A-ZŽĆČŠĐ][a-zšđčćž]{2,15}(\s[a-zšđčćž]{2,15})*$/;
    if (!reCategory.test(category.val())) {
        category.css("border", "1px solid red");
        errors.push("Ime kategorije ne sme biti kraće od tri karaktera i duže od 15!
<br/> Dozvoljena su samo slova");
    }
    else {
        category.css("border", "");
    }
    if (errors.length > 0) {
        feedbackError1.css("display", "none");
        var error = "";
        for (var x in errors) {
            error += "<p class='false er'>" + errors[x] + "</p>";
        }
        div.html(error);
    }
    else {
        $.ajaxSetup({
            headers:{
                "X-CSRF-TOKEN" : $('meta[name="_token"]').attr("content")
            }
        });
        console.log(category.val());
        $.ajax({
            url: "/admin/zanrovi/unos",
            method: "POST",
            data: {
                category: category.val()
            },
            success: function (data, xhr) {
                feedbackError1.css("display", "none");
                feeedbackOk.css("display", "block");
            },
            error: function (xhr, status, error) {
                switch (xhr.status) {
                    case 422:
                        div.html("<p class='false er'>Ime kategorije vec postoji!</p>");
                        feeedbackOk.css("display","none");
                        feedbackError1.css("display","none");
                        break;
                    case 500 :
                        feeedbackOk.css("display", "none");
                        feedbackError1.css("display", "block");
                        feedbackError.css("display", "none");
                        div.css("display", "none");
                        break;
                }
            }
        });
    }
});
```

adminInsertRole.js

```
var btn = $("#btn");
var first = $("#ime");
var last = $("#prezime");
var feeedbackOk = $("#f200");
var feedbackError1 = $("#f500");
var div = $("#errors");
var er = $(".er");
var falsee = $(".false");
var errors = [];
btn.click(() => {
    var reFirst= /^([A-ZŠĐČĆŽ][a-zšđčćž\s]{2,15})(\s[A-ZŠĐČĆŽ][a-zšđčćž\s]{2,15})*$/;
    var reLast=  /^([A-ZŠĐČĆŽ][a-zšđčćž\s]{2,25})(\s[A-ZŠĐČĆŽ][a-zšđčćž\s]{2,25})*$/;
    if(!reFirst.test(first.val().trim())){
        first.css("border","1px solid red");
        errors.push("Ime mora početi velikim slovom! <br/> Ime ne sme biti kraće od 3 i duže od 16 karaktera!");
        console.log(first.val());
    }
    else{
        first.css("border","");
    }
    if(!reLast.test(last.val().trim())){
        last.css("border","1px solid red");
        errors.push("Prezime mora početi velikim slovom!<br/>Prezime ne sme biti kraće od 3 i duže od 26 karaktera!");
    }
    else{
        last.css("border","");
    }
    if (errors.length > 0) {
        feedbackError1.css("display", "none");
        var error = "";
        for (var x in errors) {
            error += "<p class='false er'>" + errors[x] + "</p>";
        }
        div.html(error);
    }
    else {
        div.html("");
        $.ajaxSetup({
            headers:{
                "X-CSRF-TOKEN" : $('meta[name="_token"]').attr("content")
            }
        });
        $.ajax({
            url: "/admin/uloge/unos",
            method: "POST",
            data: {
                first: first.val(),
                last: last.val()
            },
            success: function (data, xhr) {
                feedbackError1.css("display", "none");
                feeedbackOk.css("display", "block");
            },
            error: function (xhr, status, error) {
                switch(xhr.status){
                    case 500:
                        feeedbackOk.css("display","none");
                        feedbackError1.css("display","block");
                        break;
                    case 422:
                        div.html("<p class='false er'>Vec postoji glumac sa ovim 
podacima!</p>");
                        feeedbackOk.css("display","none");
                        feedbackError1.css("display","none");
                        break;
                }
            }
        });
    }
});
```

adminInsertUser.js

```
var first = $("#first");
var last = $("#last");
var email = $("#email");
var btn = $("#btn");
var city = $("#city");
var roll = $("#roll");
var active = $("#active");
var username = $("#username");
var password = $("#password");
var passwordConfirm = $("#password1");
var feeedbackOk = $("#f200");
var feedbackError1 = $("#f500");
var div = $("#errors");
var er = $(".er");
var falsee = $(".false");
btn.click(() =>{
    var reFirst= /^([A-ZŠĐČĆŽ][a-zšđčćž\-']{2,15})(\s[A-ZŠĐČĆŽ][a-zšđčćž\-
']{2,15})*$/;
    var reLast=  /^([A-ZŠĐČĆŽ][a-zšđčćž\-']{2,25})(\s[A-ZŠĐČĆŽ][a-zšđčćž\-
']{2,25})*$/;
    var reEmail = /^[^@\s]{3,25}@[^@\s]{2,10}\.[^@\s]{2,7}$/;
    var cityValue = city.val();
    var rollValue = roll.val();
    var activeValue = active.val();
    var gender = $("input[type='radio']:checked");
    var reUsername = /^[a-zšđčćž0-9]{6,15}$/;
    var rePassword =  /^[A-ZŠĐČĆŽa-zšđčćž?!&^#|$%@*\/0-9]{8,15}$/;
    var errors = [];
    if(!reFirst.test(first.val())){
        first.css("border","1px solid red");
        errors.push("Ime mora početi velikim slovom! <br/> Ime ne sme biti kraće od
3 i duže od 16 karaktera!");
        console.log(first.val());
    }
    else{
        first.css("border","");
    }
    if(!reLast.test(last.val())){
        last.css("border","1px solid red");
        errors.push("Prezime mora početi velikim slovom!<br/>Prezime ne sme biti
kraće od 3 i duže od 26 karaktera!");
    }
    else{
        last.css("border","");
    }
    if(!reEmail.test(email.val())){
        email.css("border","1px solid red");
        errors.push("Email nije u dobrom formatu!");
    }
    else{
        email.css("border","");
    }
    if(cityValue==="0"){
        city.css("border","1px solid red");
        errors.push("Morate izabrati grad!");
    }
    else{
        city.css("border","");
    }
    if(!reUsername.test(username.val())){
        username.css("border","1px solid red");
        errors.push("Korisničko ime ne sme biti kraće od 6 i duže od 15
karaktera!<br/>Nisu dozvoljeni specijalni karakteri i velika slova!");
    }
    else{
        username.css("border","");
    }
    if(!rePassword.test(password.val())){
        password.css("border","1px solid red");
        errors.push("Lozinka ne sme biti kraća od 8 i duža od 15 karaktera!<br/>
Dozvoljeni su specijalni karakteri i velika slova!");
    }
    else{
        password.css("border","");
    }
    if(password.val() !== passwordConfirm.val()){
        passwordConfirm.css("border","1px solid red");
        errors.push("Lozinke se ne podudaraju!");
    }
    else if(passwordConfirm.val()===""){
        passwordConfirm.css("border","1px solid red");
        errors.push("Morate potvrditi lozinku!");
    }
    else{
        passwordConfirm.css("border","");
    }

    if(rollValue==="0"){
        roll.css("border","1px solid red");
        errors.push("Morate izabrati ulogu!");
    }
    else{
        roll.css("border","");
    }

    if(activeValue==="null"){
        active.css("border","1px solid red");
        errors.push("Morate izabrati aktivnost!");
```

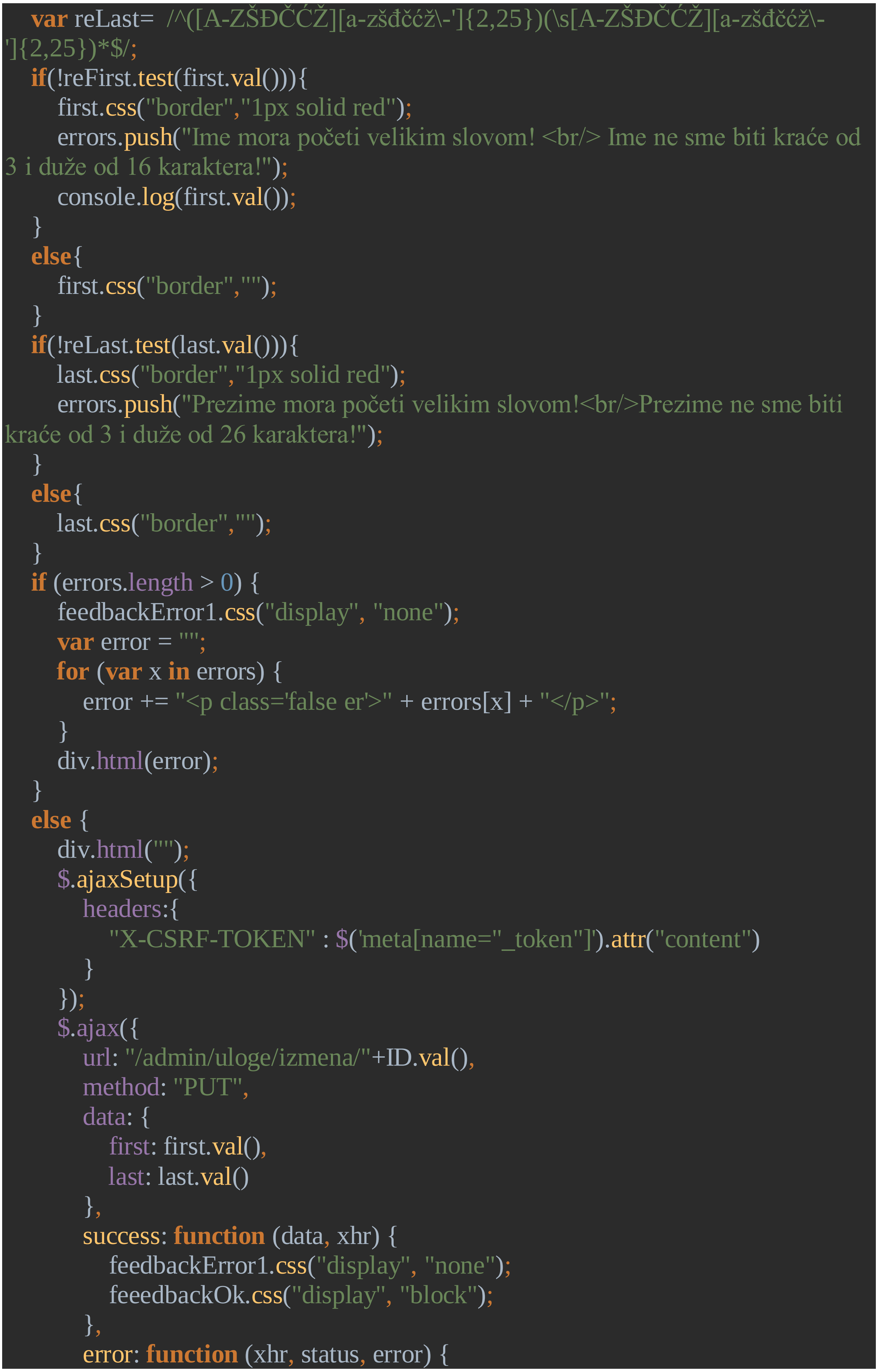
```
}
else{
    active.css("border","");
}
if(errors.length > 0){
    feedbackError1.css("display", "none");
    var error ="";
    for(var x in errors){
        error += "<p class='false er'>"+errors[x]+"</p>";
    }
    div.html(error);
}
else{
    $.ajaxSetup({
        headers:{
            "X-CSRF-TOKEN" : $('meta[name="_token"]').attr('content'),
            accepts : "application/json"
        }
    });
    div.html("");
    $.ajax({
        url:"/admin/korisnici/unos",
        type:"POST",
        dataType: "JSON",
        data:{
            firstname:first.val(),
            lastname:last.val(),
            email:email.val(),
            gender:gender.val(),
            city:cityValue,
            username:username.val(),
            password:password.val(),
            passwordConfirm:passwordConfirm.val(),
            roll:rollValue,
            active:activeValue
        },
        success:function (data,xhr) {
            feeedbackOk.css("display","block");
        },
        error:function (xhr,status,error) {
            switch(xhr.status){
                case 500:
                    feeedbackOk.css("display","none");
                    feedbackError1.css("display","block");
                    break;
                case 422:
                        div.html("<p class='false er'>Korisničko ime je zauzeto!</p>");
                        feeedbackOk.css("display","none");
                        feedbackError1.css("display","none");
                        break;
                }
            }
        });
    }
});
```

adminUpdateCategory.js

```
var btn = $("#btn");
var category = $("#category");
var feeedbackOk = $("#f200");
var feedbackError1 = $("#f500");
var div = $("#errors");
var er = $(".er");
var falsee = $(".false");
var errors = [];
var ID = $("#ID");
btn.click(() => {
    var reCategory = /^[A-ZŽĆČŠĐ][a-zšđčćž]{2,15}(\s[a-zšđčćž]{2,15})*$/;
    if (!reCategory.test(category.val())) {
        category.css("border", "1px solid red");
        errors.push("Ime kategorije ne sme biti kraće od tri karaktera i duže od 15! <br/> Dozvoljena su samo slova");
    }
    else {
        category.css("border", "");
    }
    if (errors.length > 0) {
        feedbackError1.css("display", "none");
        var error = "";
        for (var x in errors) {
            error += "<p class='false er'>" + errors[x] + "</p>";
        }
        div.html(error);
    }
    else {
        $.ajaxSetup({
            headers:{
                "X-CSRF-TOKEN" : $('meta[name="_token"]').attr("content")
            }
        });
        console.log(category.val());
        $.ajax({
            url: "/admin/zanrovi/izmena/"+ID.val(),
            method: "POST",
            data: {
                category: category.val()
            },
            success: function (data, xhr) {
                feedbackError1.css("display", "none");
                feeedbackOk.css("display", "block");
            },
            error: function (xhr, status, error) {
                switch (xhr.status) {
                    case 422:
                        div.html("<p class='false er'>Ime kategorije vec postoji!</p>");
                        feeedbackOk.css("display","none");
                        feedbackError1.css("display","none");
                        break;
                    case 500 :
                        feeedbackOk.css("display", "none");
                        feedbackError1.css("display", "block");
                        feedbackError.css("display", "none");
                        div.css("display", "none");
                        break;
                }
            }
        });
    }
});
```

adminUpdateRole.js

```
var btn = $("#btn");
var first = $("#ime");
var last = $("#prezime");
var ID = $("#ID");
var feeedbackOk = $("#f200");
var feedbackError1 = $("#f500");
var div = $("#errors");
var er = $(".er");
var falsee = $(".false");
var errors = [];
btn.click(() => {
    var reFirst= /^([A-ZŠĐČĆŽ][a-zšđčćž\-']{2,15})(\s[A-ZŠĐČĆŽ][a-zšđčćž\-']{2,15})*$/;
```



```
  var reLast= /^([A-ZŠĐČĆŽ][a-zšđčćž\-']{2,25})(\s[A-ZŠĐČĆŽ][a-zšđčćž\-
']{2,25})*$/;
  if(!reFirst.test(first.val())){
    first.css("border","1px solid red");
    errors.push("Ime mora početi velikim slovom! <br/> Ime ne sme biti kraće od
3 i duže od 16 karaktera!");
    console.log(first.val());
  }
  else{
    first.css("border","");
  }
  if(!reLast.test(last.val())){
    last.css("border","1px solid red");
    errors.push("Prezime mora početi velikim slovom!<br/>Prezime ne sme biti
kraće od 3 i duže od 26 karaktera!");
  }
  else{
    last.css("border","");
  }
  if (errors.length > 0) {
    feedbackError1.css("display", "none");
    var error = "";
    for (var x in errors) {
      error += "<p class='false er'>" + errors[x] + "</p>";
    }
    div.html(error);
  }
  else {
    div.html("");
    $.ajaxSetup({
      headers:{
        "X-CSRF-TOKEN" : $('meta[name="_token"]').attr("content")
      }
    });
    $.ajax({
      url: "/admin/uloge/izmena/"+ID.val(),
      method: "PUT",
      data: {
        first: first.val(),
        last: last.val()
      },
      success: function (data, xhr) {
        feedbackError1.css("display", "none");
        feeedbackOk.css("display", "block");
      },
      error: function (xhr, status, error) {
```

```
                alert("Doslo je do greske!");
            }
        });
    }
});
```

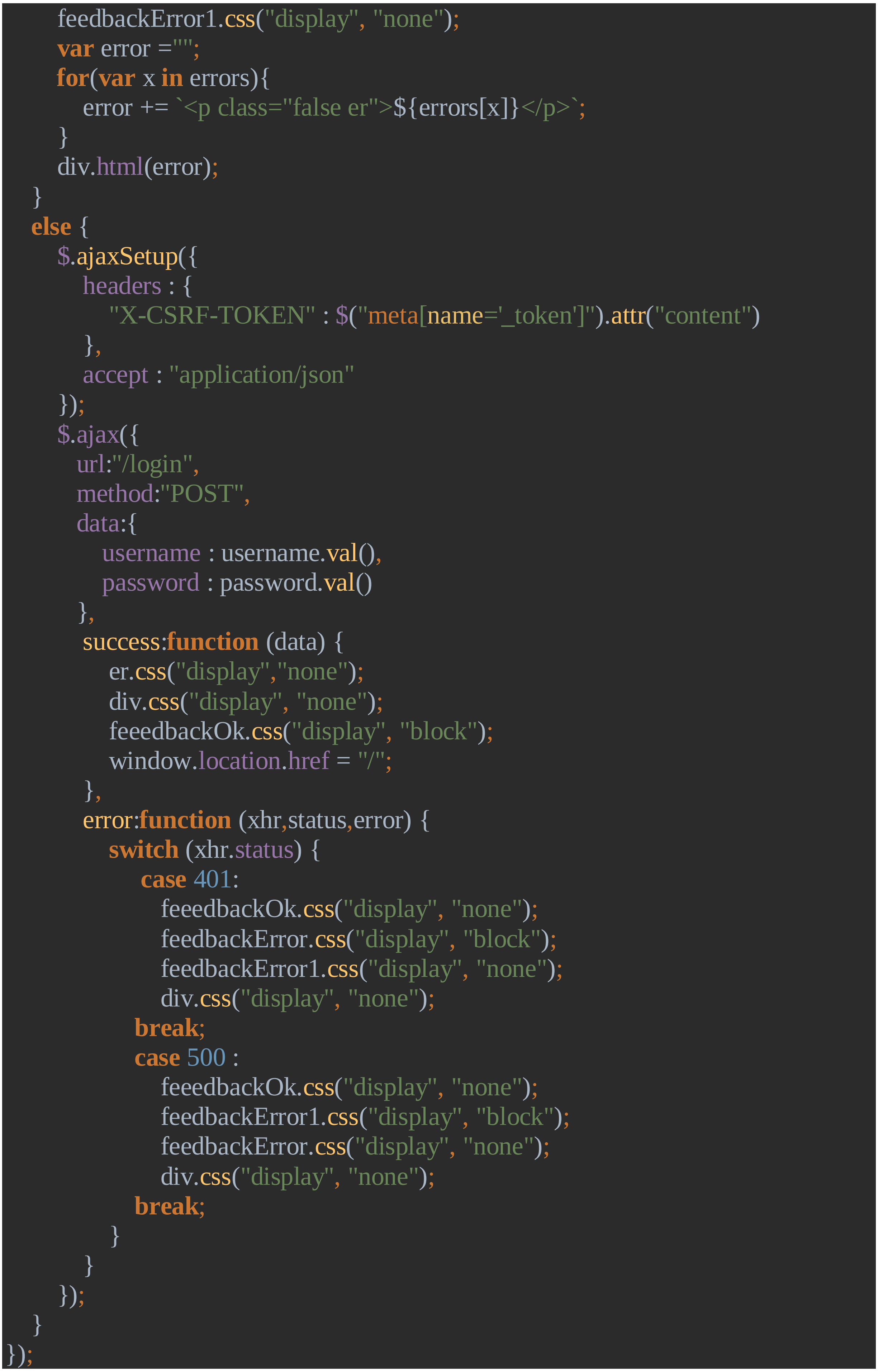
adminUpdateUser.js

```
var ID = $("#ID");
var first = $("#first");
var last = $("#last");
var email = $("#email");
var btn = $("#btn");
var city = $("#city");
var roll = $("#roll");
var active = $("#active");
var status = $("#a");
var username = $("#username");
var feeedbackOk = $("#f200");
var feedbackError1 = $("#f500");
var div = $("#errors");
var er = $(".er");
var falsee = $(".false");
btn.click(() =>{
    var reFirst= /^([A-ZŠĐČĆŽ][a-zšđčćž\-']{2,15})(\s[A-ZŠĐČĆŽ][a-zšđčćž\-']{2,15})*$/;
    var reLast=  /^([A-ZŠĐČĆŽ][a-zšđčćž\-']{2,25})(\s[A-ZŠĐČĆŽ][a-zšđčćž\-']{2,25})*$/;
    var reEmail = /^[^@\s]{3,25}@[^@\s]{2,10}\.[^@\s]{2,7}$/;
    var cityValue = city.val();
    var rollValue = roll.val();
    var activeValue = active.val();
    var gender = $("input[type='radio']:checked");
    var reUsername = /^[a-zšđčćž0-9]{6,15}$/;
    var errors = [];
    if(!reFirst.test(first.val())){
        first.css("border","1px solid red");
        errors.push("Ime mora početi velikim slovom! <br/> Ime ne sme biti kraće od 3 i duže od 16 karaktera!");
        console.log(first.val());
    }
    else{
        first.css("border","");
    }
    if(!reLast.test(last.val())){
        last.css("border","1px solid red");
        errors.push("Prezime mora početi velikim slovom!<br/>Prezime ne sme biti kraće od 3 i duže od 26 karaktera!");
    }
    else{
        last.css("border","");
    }
    if(!reEmail.test(email.val())){
        email.css("border","1px solid red");
        errors.push("Email nije u dobrom formatu!");
    }
    else{
        email.css("border","");
    }
    if(cityValue==="0"){
        city.css("border","1px solid red");
        errors.push("Morate izabrati grad!");
    }
    else{
        city.css("border","");
    }
    if(!reUsername.test(username.val())){
        username.css("border","1px solid red");
        errors.push("Korisničko ime ne sme biti kraće od 6 i duže od 15 karaktera!<br/>Nisu dozvoljeni specijalni karakteri i velika slova!");
    }
    else{
        username.css("border","");
    }
    if(rollValue==="0"){
        roll.css("border","1px solid red");
        errors.push("Morate izabrati ulogu!");
    }
    else{
        roll.css("border","");
    }

    if(activeValue==="null"){
        active.css("border","1px solid red");
        errors.push("Morate izabrati aktivnost!");
    }
    else{
        active.css("border","");
    }
    if($('#a').val()==="null"){
        $('#a').css("border","1px solid red");
        errors.push("Morate izabrati aktivnost!");
    }
    else{
        $('#a').css("border","");
    }
    if(errors.length > 0){
        feedbackError1.css("display", "none");
        var error ="";
        for(var x in errors){
            error += "<p class='false er'>"+errors[x]+"</p>";
        }
        div.html(error);
    }
    else{
        $.ajaxSetup({
            headers : {
                "X-CSRF-TOKEN" : $("meta[name='_token']").attr("content")
            }
        });
        $.ajax({
            url:"/admin/korisnici/izmena/"+ID.val(),
            method:"POST",
            data:{
                ID:ID.val(),
                firstname:first.val(),
                lastname:last.val(),
                email:email.val(),
                gender:gender.val(),
                city:cityValue,
                username:username.val(),
                roll:rollValue,
                active:activeValue,
                status:$("#a").val(),
            },
            success:function (data,xhr) {
                feeedbackOk.css("display","block");
            },
            error:function (xhr,status,error) {
                switch (xhr.status) {
                    case 404:
                        window.location.href="../greska.php";
                        break;
                    case 422:
                        alert("Izvinjavamo se zbog tehničkih problema");
                        break;
                    case 409:
```

```
                    div.html("<p class='false er'>Korisničko ime je zauzeto!</p>");
                    feeedbackOk.css("display","none");
                    feedbackError1.css("display","none");
                    break;

                }
            }
        });
    }
});
```

login.js

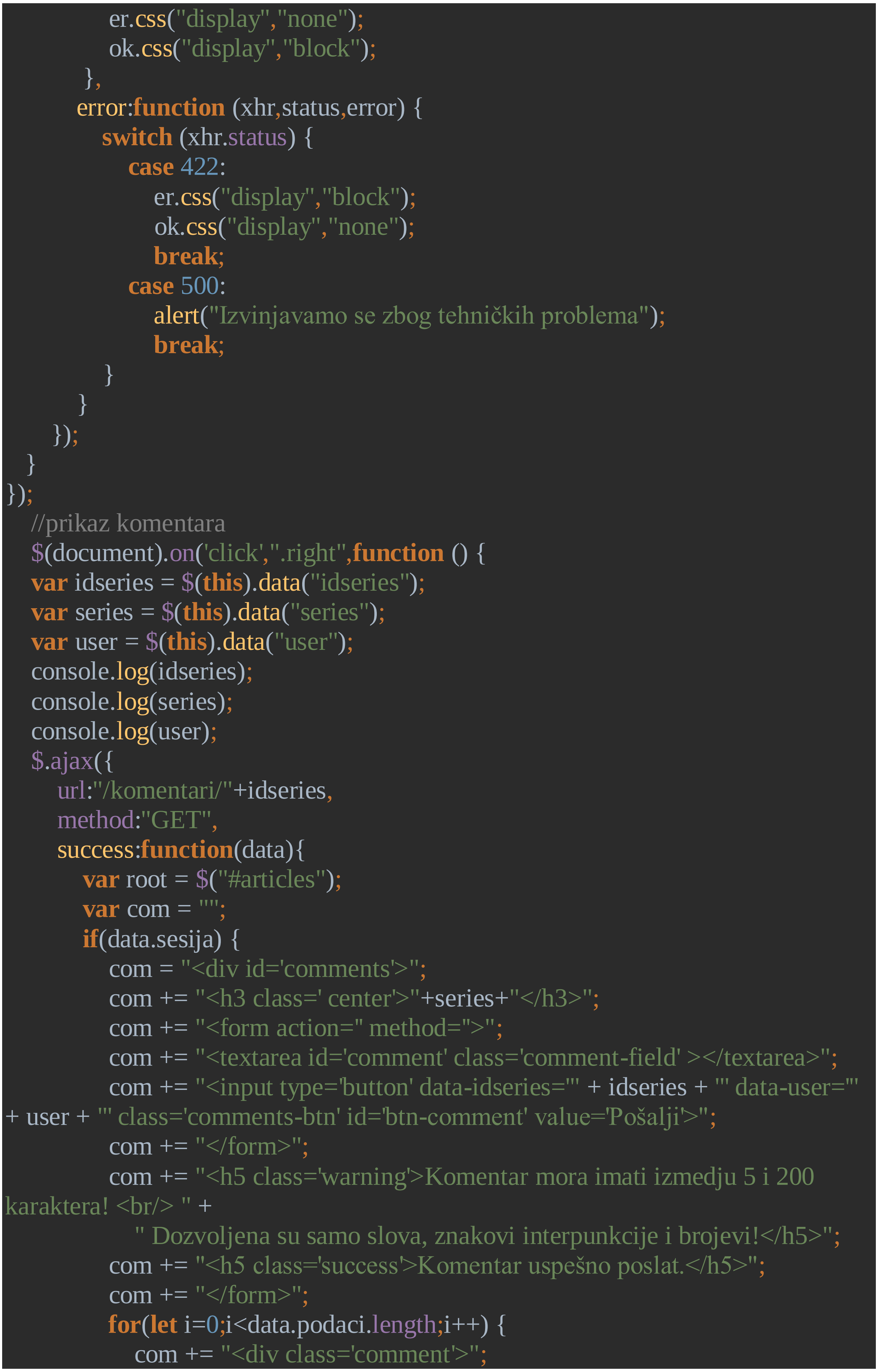
```

var username = $("#username");
var password = $("#password");
var btn = $("#btn");
var feeedbackOk = $("#f200");
var feedbackError = $("#f400");
var feedbackError1 = $("#f500");
var div = $("#errors");
var er = $(".er");
var falsee = $(".false");
btn.click(function () {
    var reUsername = /^[a-zšđčćž0-9]{6,15}$/;
    var rePassword =  /^[A-ZŠĐČĆŽa-zšđčćž?!&^#|$%@*\/0-9]{8,15}$/;
    var errors = [];
   if(!reUsername.test(username.val())){
       username.css("border","1px solid red");
       errors.push("Korisničko ime ne sme biti kraće od 6 i duže od 15 karaktera!
<br/> Nisu dozvoljeni specijalni karakteri i velika slova!");
   }
   else{
       username.css("border","");
   }
    if(!rePassword.test(password.val())){
        password.css("border","1px solid red");
        errors.push("Lozinka ne sme biti kraća od 8 i duža od 15 karaktera!<br/>
Dozvoljeni su specijalni karakteri i velika slova!");
    }
    else{
        password.css("border","");
    }
    if(errors.length > 0){
        feedbackError.css("display", "none");
```



```
        feedbackError1.css("display", "none");
        var error ="";
        for(var x in errors){
            error += `<p class="false er">${errors[x]}</p>`;
        }
        div.html(error);
    }
    else {
        $.ajaxSetup({
            headers : {
                "X-CSRF-TOKEN" : $("meta[name='_token']").attr("content")
            },
            accept : "application/json"
        });
        $.ajax({
           url:"/login",
           method:"POST",
           data:{
               username : username.val(),
               password : password.val()
           },
            success:function (data) {
                er.css("display","none");
                div.css("display", "none");
                feeedbackOk.css("display", "block");
                window.location.href = "/";
            },
            error:function (xhr,status,error) {
                switch (xhr.status) {
                     case 401:
                        feeedbackOk.css("display", "none");
                        feedbackError.css("display", "block");
                        feedbackError1.css("display", "none");
                        div.css("display", "none");
                    break;
                    case 500 :
                        feeedbackOk.css("display", "none");
                        feedbackError1.css("display", "block");
                        feedbackError.css("display", "none");
                        div.css("display", "none");
                    break;
                }
            }
        });
    }
});
```

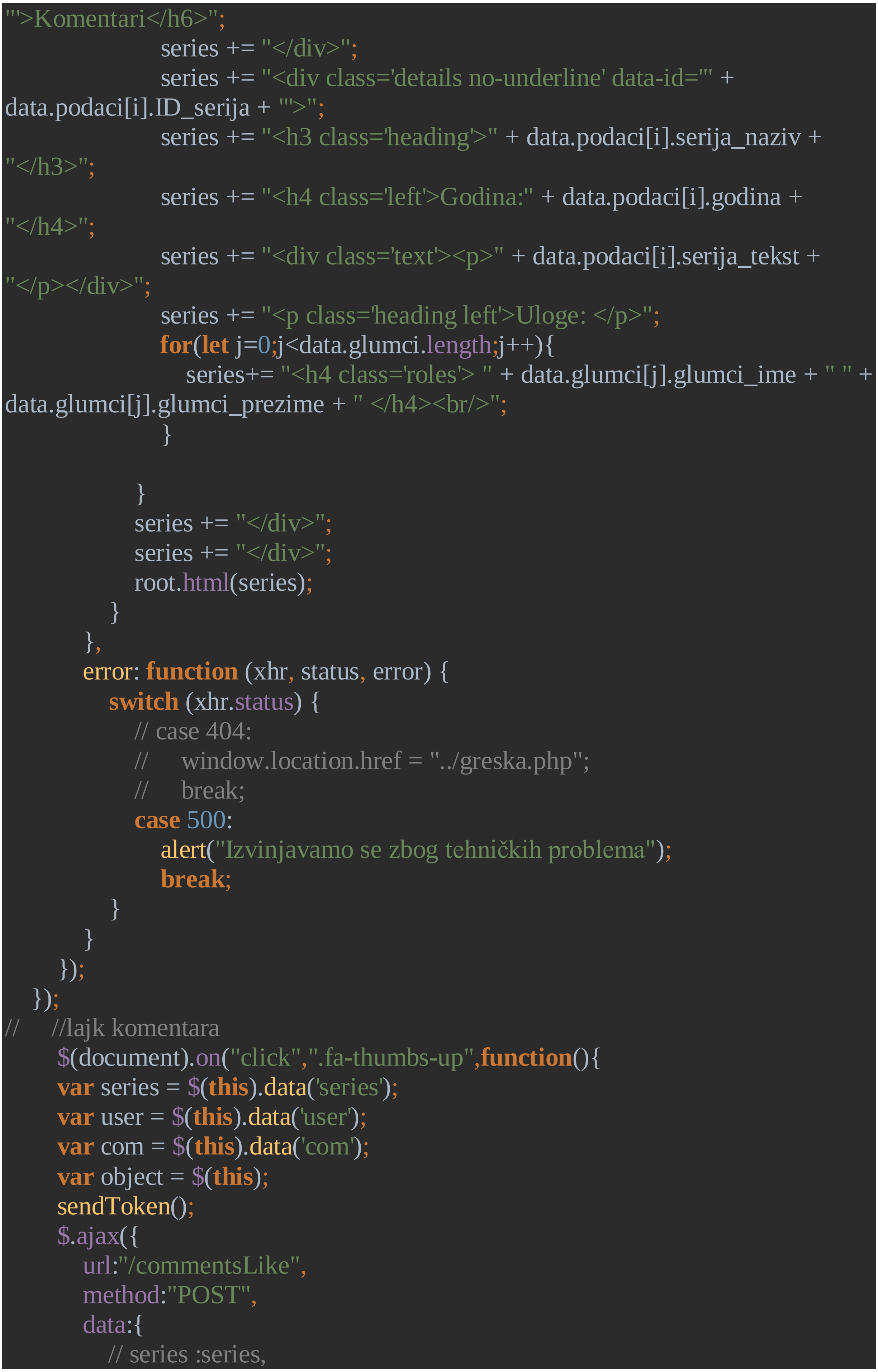
main.js

```
function sendToken() {
    $.ajaxSetup({
        headers : {
            "X-CSRF-TOKEN" : $("meta[name='_token']").attr("content")
        }
        // accepts: "application/json"
    });
}
$(document).ready(function() {
//vracanje na prijavu
    $(document).on("click",".redirect",function () {
        window.location.href = '/prijava';
    });
//vracanje na serije
    $(document).on('click','.refresh',function (){
       location.reload();
    });
//     //slanje komenatara
   $(document).on("click","#btn-comment",function () {
   var user = $(this).data("user");
   var series = $(this).data("idseries");
   var commentValue = $('.comment-field').val();
   var errors = [];
   var er = $(".warning");
   var ok = $(".success");
   var regex = /^[a-zšđčćž0-9A-ZŠĐČĆŽ?!\.\s]{5,200}$/;
   if(!regex.test(commentValue)){
            errors.push("Greska");
   }
   if(errors.length > 0) {
       ok.css("display","none");
       er.css("display","block");
   }
   else{
       sendToken();
       $.ajax({
            url:"/komentari",
            method:"POST",
            data:{
                iduser : user,
                idseries : series,
                comment : commentValue
            },
            success:function (data) {
```

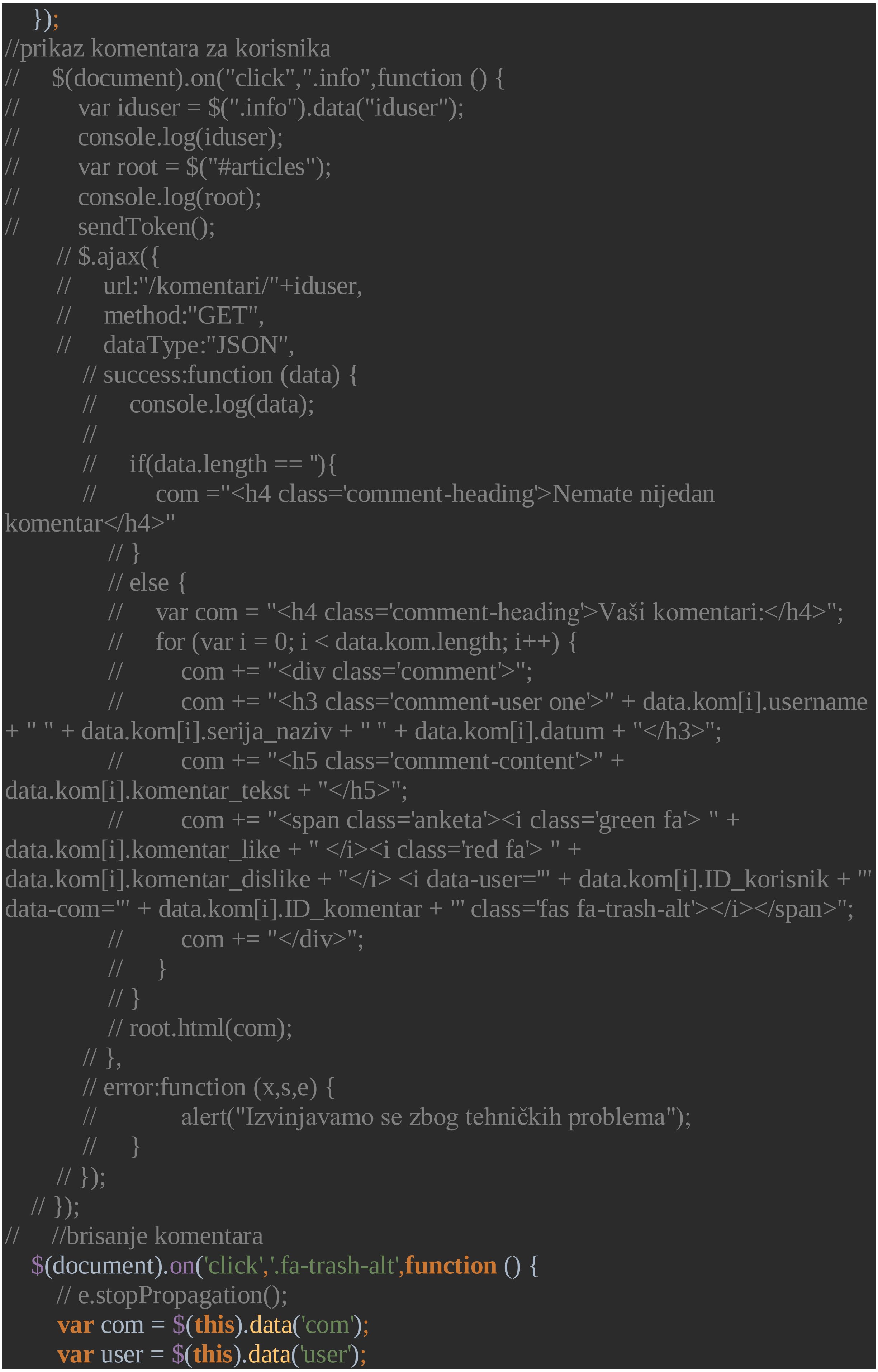


```
                er.css("display","none");
                ok.css("display","block");
            },
           error:function (xhr,status,error) {
               switch (xhr.status) {
                   case 422:
                       er.css("display","block");
                       ok.css("display","none");
                       break;
                   case 500:
                       alert("Izvinjavamo se zbog tehničkih problema");
                       break;
               }
           }
       });
   }
});
    //prikaz komentara
    $(document).on('click','.right',function () {
    var idseries = $(this).data("idseries");
    var series = $(this).data("series");
    var user = $(this).data("user");
    console.log(idseries);
    console.log(series);
    console.log(user);
    $.ajax({
        url:"/komentari/"+idseries,
        method:"GET",
        success:function(data){
            var root = $("#articles");
            var com = "";
            if(data.sesija) {
                com = "<div id='comments'>";
                com += "<h3 class=' center'>"+series+"</h3>";
                com += "<form action='' method=''>";
                com += "<textarea id='comment' class='comment-field' ></textarea>";
                com += "<input type='button' data-idseries='" + idseries + "' data-user='"
+ user + "' class='comments-btn' id='btn-comment' value='Pošalji'>";
                com += "</form>";
                com += "<h5 class='warning'>Komentar mora imati izmedju 5 i 200
karaktera! <br/> " +
                    " Dozvoljena su samo slova, znakovi interpunkcije i brojevi!</h5>";
                com += "<h5 class='success'>Komentar uspešno poslat.</h5>";
                com += "</form>";
                for(let i=0;i<data.podaci.length;i++) {
                    com += "<div class='comment'>";
                    com += "<h3 class='comment-user'>" + data.podaci[i].username +" " +data.podaci[i].datum+ "</h3>";
                    com += "<h5 class='comment-content'>" + data.podaci[i].komentar_tekst + "</h5>";
                    if(data.podaci[i].komentar_tekst != null)
                    com += "<span class='anketa'>";
                    if(data.podaci[i].komentator == user) {
                        console.log(data.podaci[i].komentator)
                        com += "<i class='green fa'> " + data.podaci[i].komentar_like + " </i><i class='red fa'> " + data.podaci[i].komentar_dislike + "</i> <i data-user='" + data.podaci[i].ID_korisnik + "' data-com='" + data.podaci[i].ID_komentar + "' class='fas fa-trash-alt'></i></span>";
                    }
                    else {
                        com += "<i class='fa fa-thumbs-up' data-com='" + data.podaci[i].ID_komentar + "' data-user='" + user + "' data-series='" + data.podaci[i].ID_vest + "'> " + data.podaci[i].komentar_like + " </i>  <i class='fa fa-thumbs-down' data-user='" + user + "' data-idseries='" + data.podaci[i].ID_serija + "' data-com='" + data.podaci[i].ID_komentar + "'> ";
                        com += data.podaci[i].komentar_dislike + "</i></span>";
                    }
                    com += "</div>";
                }
                com += "<h6 class='refresh'>Povratak na serije</h6>";
                root.html(com);
            }
            else{
                com = "<div id='comments'>";
                for(let i=0;i<data.podaci.length;i++) {
                    com += "<div class='comment'>";
                    com += "<h3 class='comment-user'>" + data.podaci[i].username +" " +data.podaci[i].datum+ "</h3>";
                    com += "<h5 class='comment-content'>" + data.podaci[i].komentar_tekst + "</h5>";
                    com += "<span class='anketa'>";
                    com += "<i class='green fa'> " + data.podaci[i].komentar_like + " </i><i class='red fa'> " + data.podaci[i].komentar_dislike + "</i>";
                    com += "</div>";
                }
                com += "</div><h5 class='forbidden'><a class='redirect' >Morate se ulogovati da biste komentarisali</a></h5>";
                com += "<h6 class='refresh'>Povratak na serije</h6>";
                root.html(com);
            }
        },
        error:function (xhr, status, error) {
```

```
                    alert("Izvinjavamo se zbog tehničkih problema");

        }
    });
});
// pretraga serije
    $(document).on('keydown','#search',function (){
    var char = $('#search').val();
    console.log(char);
    $.ajax({
        url: "/pretraga/"+char,
        method: "GET",
        success: function (data) {
            var root = $("#articles");
            var series = "";
            var tekst = "";
            if(data.korisnik) {
                for (let i = 0; i < data.podaci.length; i++) {
                    series += "<div class='item'>";
                    tekst = data.podaci[i].serija_tekst;
                    series += "<div class='img'>" +
                        "<img src='" + data.podaci[i].slika + "' alt='" +
data.podaci[i].serija_naziv + "' title='" + data.podaci[i].serija_naziv + "' width='200'
height='150'>" +
                        "</div>";
                    series += "<div class='right-container'>";
                    series += "<div class='date-com'>";
                    series += "<i class='left vote-series-plus fa fa-plus' data-idseries='" +
data.podaci[i].ID_serija + "' data-user='" + data.korisnik.ID_korisnik + "' data-
series='" + data.podaci[i].serija_naziv + "' > " + data.podaci[i].serija_like+ " </i>";
                    series += "<i class='left vote-series-plus fa fa-minus' data-idseries='" +
data.podaci[i].ID_serija + "' data-user='" + data.korisnik.ID_korisnik + "' data-
series='" + data.podaci[i].serija_naziv + "' > " + data.podaci[i].serija_dislike+ "
</i>";
                    series += "<h6 class='right' data-idseries='" + data.podaci[i].ID_serija
+ "' data-user='" + data.korisnik.ID_korisnik + "' data-series='" +
data.podaci[i].serija_naziv + "'>Komentari</h6>";
                    series += "</div>";
                    series += "<div class='details' data-idseries='" +
data.podaci[i].ID_serija + "'><h3 class='heading'>" + data.podaci[i].serija_naziv +
"</h3><p class='text'>" + tekst.substring(0, 150) + "...</p></div>";
                    series += "</div>";
                    series += "</div>";
                }
                root.html(series);
            }
            else{

                for (let i = 0; i < data.podaci.length; i++) {
                    series += "<div class='item'>";

                    tekst = data.podaci[i].serija_tekst;
                    series += "<div class='img'>" +
                        "<img src='" + data.podaci[i].slika + "' alt='" + data.podaci[i].serija_naziv + "' title='" + data.podaci[i].serija_naziv + "' width='200' height='150'>" +
                        "</div>";
                    series += "<div class='right-container'>";
                    series += "<div class='date-com'>";
                    series +="<a class='redirect' ><i class='left fa fa-plus'></i><i class='left fa fa-minus'></i></a>";
                    series += "<h6 class='right' data-idseries='" + data.podaci[i].ID_serija + "' data-series='" + data.podaci[i].serija_naziv + "' >Komentari</h6>";
                    series += "</div>";
                    series += "<div class='details' data-idseries='" + data.podaci[i].ID_serija + "'><h3 class='heading'>" + data.podaci[i].serija_naziv + "</h3><p class='text'>" + tekst.substring(0, 150) + "...</p></div>";
                    series += "</div>";
                    series += "</div>";
                }
                root.html(series);
            }
        },
        error: function (xhr, status, error) {

        }
    });
});
    //prikaz jedne serije
        $(document).on('click','.details',function () {
        var user = $(this).data("user");
        var series = $(this).data("series");
        var id = $(this).data("idseries");
        console.log(id);
        //ovo radimo
        sendToken();
        $.ajax({
            url: "/serija/"+id,
            method: "GET",
            dataType:"JSON",
            success: function (data) {
                var root = $("#articles");
        var series = "<div class='news'>";
        console.log(data.podaci);
        console.log(data.podaci.uloge);
        if(data.korisnik) {
          for (let i = 0; i < data.podaci.length; i++) {
            series += "<div>" +
              "<img src='" + data.podaci[i].slika + "' alt='" + data.podaci[i].serija_naziv + "' title='" + data.podaci[i].serija_naziv + "' width='100%' height='300px'>" +
              "</div>";
            series += "<div class='right-container'>";
            series += "<div class='date-com'>";
            series += "<h6 class='right' data-idseries='" + data.podaci[i].ID_serija + "' data-user='" + data.korisnik.ID_korisnik + "' data-series='" + data.podaci[i].serija_naziv + "'>Komentari</h6>";
            series += "</div>";
            series += "<div class='details no-underline' data-id='" + data.podaci[i].ID_serija + "'>";
            series += "<h3 class='heading'>" + data.podaci[i].serija_naziv + "</h3>";
            series += "<h4 class='left'>Godina:" + data.podaci[i].godina + "</h4>";
            series += "<div class='text'><p>" + data.podaci[i].serija_tekst + "</p></div>";
            series += "<p class='heading left'>Uloge: </p>";
            for(let j=0;j<data.glumci.length;j++){
              series+= "<h4 class='roles'> " + data.glumci[j].glumci_ime + " " + data.glumci[j].glumci_prezime + " </h4><br/>";
            }

            series += "</div>";
          }
          series += "</div>";
          root.html(series);
        }
        else{
          for (let i = 0; i < data.podaci.length; i++) {
            series += "<div>" +
              "<img class='big' src='" + data.podaci[i].slika + "' alt='" + data.podaci[i].serija_naziv + "' title='" + data.podaci[i].serija_naziv + "' width='100%' height='300px'>" +
              "</div>";
            series += "<div class='right-container'>";
            series += "<div class='date-com'>";
            series += "<h6 class='right' data-idseries='" + data.podaci[i].ID_serija + "' data-series='" + data.podaci[i].serija_naziv +
```



```
"'>Komentari</h6>";
                        series += "</div>";
                        series += "<div class='details no-underline' data-id='" +
data.podaci[i].ID_serija + "'>";
                        series += "<h3 class='heading'>" + data.podaci[i].serija_naziv +
"</h3>";
                        series += "<h4 class='left'>Godina:" + data.podaci[i].godina +
"</h4>";
                        series += "<div class='text'><p>" + data.podaci[i].serija_tekst +
"</p></div>";
                        series += "<p class='heading left'>Uloge: </p>";
                        for(let j=0;j<data.glumci.length;j++){
                            series+= "<h4 class='roles'> " + data.glumci[j].glumci_ime + " " +
data.glumci[j].glumci_prezime + " </h4><br/>";
                        }

                    }
                    series += "</div>";
                    series += "</div>";
                    root.html(series);
                }
            },
            error: function (xhr, status, error) {
                switch (xhr.status) {
                    // case 404:
                    //     window.location.href = "../greska.php";
                    //     break;
                    case 500:
                        alert("Izvinjavamo se zbog tehničkih problema");
                        break;
                }
            }
        });
    });
//     //lajk komentara
        $(document).on("click",".fa-thumbs-up",function(){
        var series = $(this).data('series');
        var user = $(this).data('user');
        var com = $(this).data('com');
        var object = $(this);
        sendToken();
        $.ajax({
            url:"/commentsLike",
            method:"POST",
            data:{
                // series :series,
                user : user,
                com : com
            },
            success:function(data){
                object.html(" Vaš glas je zabeležen ");
            },
            error:function(x,s,e){
                switch (x.status) {
                    case 409:
                        object.html(" Već ste glasali! ");
                        break;
                    case 500:
                        alert("Izvinjavamo se zbog tehničkih problema");
                        break;
                }
            }
        });
    });
//dislike komentara
        $(document).on("click",".fa-thumbs-down",function(){
        var series = $(this).data('series');
        var user = $(this).data('user');
        var com = $(this).data('com');
        var object = $(this);
        sendToken();
        $.ajax({
            url:"/commentsDislike",
            method:"POST",
            data:{
                user : user,
                com : com
            },
            success:function(data){
                object.html(" Vaš glas je zabeležen ");
            },
            error:function(x,s,e){
                switch (x.status) {
                    case 409:
                        object.html(" Već ste glasali ");
                        break;
                    case 500:
                        alert("Izvinjavamo se zbog tehničkih problema");
                        break;
                }
            }
        })
```

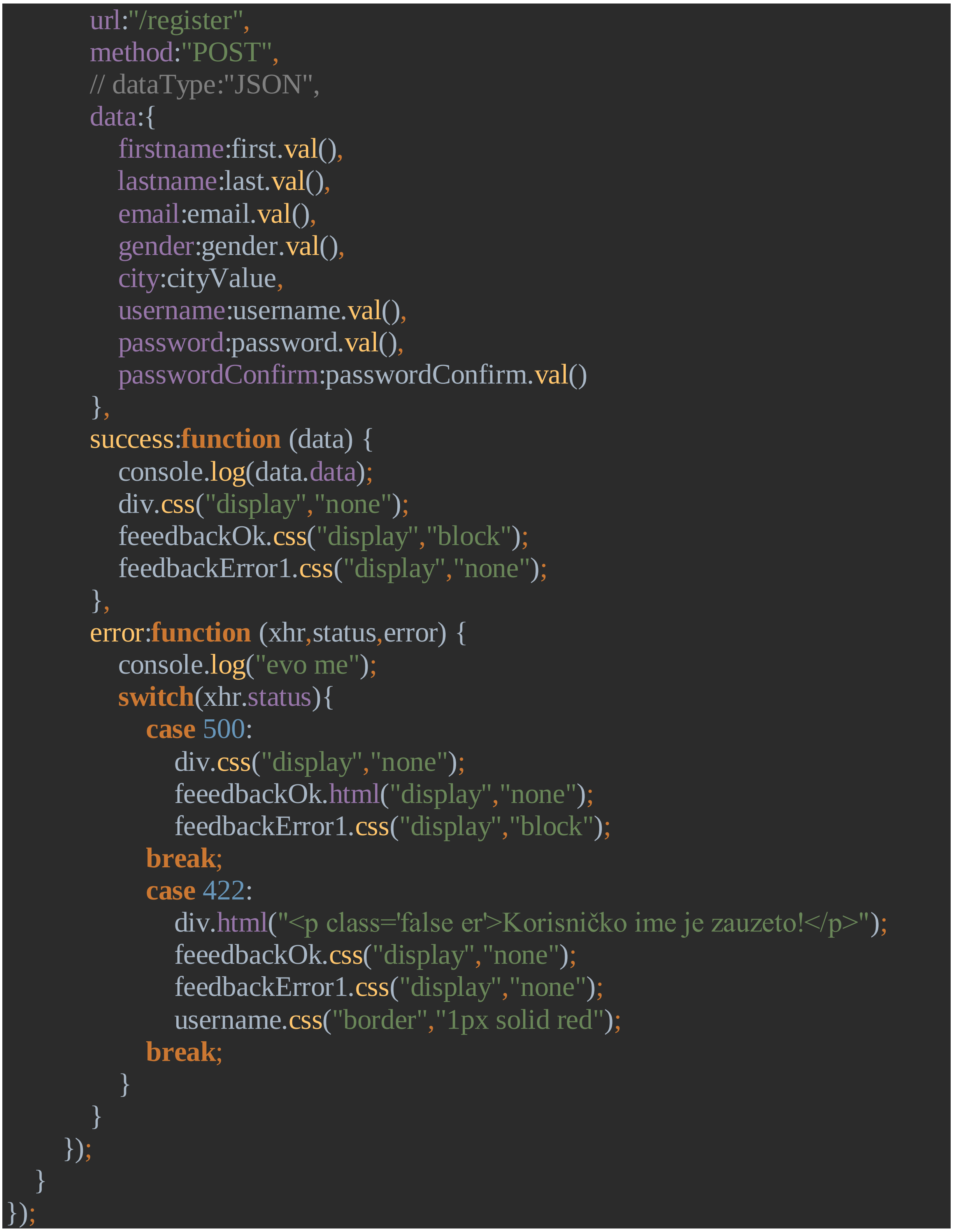


```
    });
//prikaz komentara za korisnika
//     $(document).on("click",".info",function () {
//         var iduser = $(".info").data("iduser");
//         console.log(iduser);
//         var root = $("#articles");
//         console.log(root);
//         sendToken();
        // $.ajax({
        //     url:"/komentari/"+iduser,
        //     method:"GET",
        //     dataType:"JSON",
            // success:function (data) {
            //     console.log(data);
            //
            //     if(data.length == ''){
            //         com ="<h4 class='comment-heading'>Nemate nijedan
komentar</h4>"
                // }
                // else {
                //     var com = "<h4 class='comment-heading'>Vaši komentari:</h4>";
                //     for (var i = 0; i < data.kom.length; i++) {
                //         com += "<div class='comment'>";
                //         com += "<h3 class='comment-user one'>" + data.kom[i].username
+ " " + data.kom[i].serija_naziv + " " + data.kom[i].datum + "</h3>";
                //         com += "<h5 class='comment-content'>" +
data.kom[i].komentar_tekst + "</h5>";
                //         com += "<span class='anketa'><i class='green fa'> " +
data.kom[i].komentar_like + " </i><i class='red fa'> " +
data.kom[i].komentar_dislike + "</i> <i data-user='" + data.kom[i].ID_korisnik + "'
data-com='" + data.kom[i].ID_komentar + "' class='fas fa-trash-alt'></i></span>";
                //         com += "</div>";
                //     }
                // }
                // root.html(com);
            // },
            // error:function (x,s,e) {
            //             alert("Izvinjavamo se zbog tehničkih problema");
            //     }
        // });
    // });
//     //brisanje komentara
    $(document).on('click','.fa-trash-alt',function () {
        // e.stopPropagation();
        var com = $(this).data('com');
        var user = $(this).data('user');
        sendToken();
        $.ajax({
            url:"commentsDelete",
            method:"DELETE",
            data:{
                com:com,
                user:user
            },
            success:function (data) {
                alert("Uspešno ste obrisali komentar!");
                location.reload();
            },
            error:function (x,s,e) {
                alert("Izvinjavamo se zbog tehničkih problema");
                }
        });
    });
//dislajk serije
    $(document).on("click",".vote-series-minus",function(){
        var series = $(this).data('series');
        var user = $(this).data('user');
        var object = $(this);
        console.log(series);
        console.log(user);
        sendToken();
        $.ajax({
            url:"/seriesDislike",
            method:"POST",
            data:{
                series:series,
                user:user
            },
            success:function(data){
                console.log("da");
                console.log(data);
                object.html("Vaš glas je zabeležen");
            },
            error:function(x,s,e){
                console.log("ne");
                switch (x.status) {
                    case 409:
                        object.html(" Već ste glasali ");
                        break;
                    case 500:
                        alert("Izvinjavamo se zbog tehničkih problema");
                        break;
                }
            }
        });
    });
    //lajk serije
    $(document).on("click",".vote-series-plus",function(){
        var series = $(this).data('series');
        var user = $(this).data('user');
        var object = $(this);
        console.log(series);
        console.log(user);
        sendToken();
        $.ajax({
            url:"/seriesLike",
            method:"POST",
            data:{
                series:series,
                user:user
            },
            success:function(data){
                console.log(data);
                object.html("Vaš glas je zabeležen");
            },
            error:function(x,s,e){
                console.log("ne");
                switch (x.status) {
                    case 409:
                        object.html(" Već ste glasali ");
                        break;
                    case 500:
                        alert("Izvinjavamo se zbog tehničkih problema");
                        break;
                }
            }
        });
    });
});
```

registration.js

```
var first = $("#first");
var last = $("#last");
var email = $("#email");
var btn = $("#btn");
var city = $("#city");
var username = $("#username");
var password = $("#password");
var passwordConfirm = $("#password1");
var feeedbackOk = $("#f200");
var feedbackError1 = $("#f500");
var div = $("#errors");
var er = $(".er");
var falsee = $(".false");
btn.click(function() {
   var reFirst= /^([A-ZŠĐČĆŽ][a-zšđčćž\-']{2,15})(\s[A-ZŠĐČĆŽ][a-zšđčćž\-']{2,15})*$/;
   var reLast=  /^([A-ZŠĐČĆŽ][a-zšđčćž\-']{2,25})(\s[A-ZŠĐČĆŽ][a-zšđčćž\-']{2,25})*$/;
   var reEmail = /^[^@\s]{3,25}@[^@\s]{2,10}\.[^@\s]{2,7}$/;
   var cityValue = city.val();
   var gender = $("input[type='radio']:checked");
   var reUsername = /^[a-zšđčćž0-9]{6,15}$/;
   var rePassword =  /^[A-ZŠĐČĆŽa-zšđčćž?!&^#|$%@*\/0-9]{8,15}$/;
   var errors = [];
   if(!reFirst.test(first.val())){
      first.css("border","1px solid red");
      errors.push("Ime mora početi velikim slovom! <br/> Ime ne sme biti kraće od 3 i duže od 16 karaktera!");
      console.log(first.val());
   }
   else{
      first.css("border","");
   }
   if(!reLast.test(last.val())){
      last.css("border","1px solid red");
      errors.push("Prezime mora početi velikim slovom!<br/>Prezime ne sme biti kraće od 3 i duže od 26 karaktera!");
   }
   else{
      last.css("border","");
   }
   if(!reEmail.test(email.val())){
      email.css("border","1px solid red");
      errors.push("Email nije u dobrom formatu!");
   }
   else{
      email.css("border","");
   }
   if(cityValue==="0"){
      city.css("border","1px solid red");
      errors.push("Morate izabrati grad!");
```

```
    }
    else{
        city.css("border","");
    }
    if(!reUsername.test(username.val())){
        username.css("border","1px solid red");
        errors.push("Korisničko ime ne sme biti kraće od 6 i duže od 15 karaktera!
<br/> Nisu dozvoljeni specijalni karakteri i velika slova!");
    }
    else{
        username.css("border","");
    }
    if(!rePassword.test(password.val())){
        password.css("border","1px solid red");
        errors.push("Lozinka ne sme biti kraća od 8 i duža od 15 karaktera!<br/>
Dozvoljeni su specijalni karakteri i velika slova!");
    }
    else{
        password.css("border","");
    }
    if(password.val() !== passwordConfirm.val()){
        passwordConfirm.css("border","1px solid red");
        errors.push("Lozinke se ne podudaraju!");
    }
    else if(passwordConfirm.val()===""){
        passwordConfirm.css("border","1px solid red");
        errors.push("Morate potvrditi lozinku!");
    }
    else{
        passwordConfirm.css("border","");
    }
    if(errors.length > 0){
        feedbackError1.css("display", "none");
        var error ="";
        for(var x in errors){
            error += "<p class='false er'>"+errors[x]+"</p>";
        }
        div.html(error);
    }
    else{
        $.ajaxSetup({
            headers : {
                "X-CSRF-TOKEN" : $("meta[name='_token']").attr("content")
            }
        });
        $.ajax({
```



```
            url:"/register",
            method:"POST",
            // dataType:"JSON",
            data:{
                firstname:first.val(),
                lastname:last.val(),
                email:email.val(),
                gender:gender.val(),
                city:cityValue,
                username:username.val(),
                password:password.val(),
                passwordConfirm:passwordConfirm.val()
            },
            success:function (data) {
                console.log(data.data);
                div.css("display","none");
                feeedbackOk.css("display","block");
                feedbackError1.css("display","none");
            },
            error:function (xhr,status,error) {
                console.log("evo me");
                switch(xhr.status){
                    case 500:
                        div.css("display","none");
                        feeedbackOk.html("display","none");
                        feedbackError1.css("display","block");
                    break;
                    case 422:
                        div.html("<p class='false er'>Korisničko ime je zauzeto!</p>");
                        feeedbackOk.css("display","none");
                        feedbackError1.css("display","none");
                        username.css("border","1px solid red");
                    break;
                }
            }
        });
    }
});
```

--------------------------------resources/views/layouts------------------------

form.blade.php

```
@include('partials.headForm')
@yield('content')
```

master.blade.php

```
@include('partials.header')
@include('partials.left-side')
@yield('content')
@include('partials.right-side')
@include('partials.footer')
```

-------------------------------resources/views/pages----------------------------

autor.blade.php

```
@extends('layouts.master')
@section('title','Autor')
@section('content')
<div id="articles">
    <div id="autor">
        <div id="autor_slika">
            <img src="images/ja.jpg" alt="slika-autora" width="200" height="200"
title="Dušan Krsmanović"/>
        </div>
        <div id="autor_tekst">
            <h4> Autor ovog sajta je Dušan Krsmanović 28/16, student Visoke ICT
škole,
                smera internet tehnologije. Ovaj sajt je namenjen isključivo u edukativne
svrhe kao deo predispitnih obaveza.
            </h4>
        </div>
    </div>
</div>
@endsection
```

dodaj-glumca.blade.php

```
@extends('layouts.form')
@section('title','Dodaj glumca')
```

```
<div id="wrapper">
    <div id="box">
        <div id="errors">
            <noscript>
                <p class="false er">Molimo Vas omogućite JavuScript kako biste dodali glumca!</p>
            </noscript>
        </div>
        <p class="feedback ok" id="f200">Uspešno ste dodali glumca!<br/></p>
        <div id="header">
            <h3>Dodaj glumca</h3>
        </div>
        <form id="forma2" action="" method="">
            <div id="content1" class="admin">
                <p class="left">Ime</p>
                <input type="text"  class="right" id="ime" autofocus="autofocus" placeholder="Petar"/>
                <p class="left">Prezime</p>
                <input type="text"  class="right" id="prezime"  placeholder="Petrovic"/>
                <input type="button" value="DODAJ" class="btn" id="btn"/>
            </div>
        </form>
        <footer>
            <h5>
                <a href="{{route('role-show')}}">Nazad na spisak glumaca</a>
            </h5>
        </footer>
    </div>
    <script src="{{asset("/")}}../js/jquery-3.3.1.min.js" type="text/javascript"></script>
    <script src="{{asset("/")}}../js/adminInsertRole.js" type="text/javascript"></script>
```

dodaj-kategoriju.blade.php

```
@extends('layouts.form')
@section('title','Dodaj kategoriju')
<div id="wrapper">
    <div id="box">
        <div id="errors">
            <noscript>
                <p class="false er">Molimo Vas omogućite JavuScript kako biste dodali kategoriju!</p>
            </noscript>
        </div>
        <p class="feedback ok" id="f200">Uspešno ste dodali kategoriju!<br/></p>
        <p class="feedback er" id="f500">Trenutno ne možete dodati kategoriju</p>
        <div id="header">
            <h3>Dodaj kategoriju</h3>
        </div>
        <form id="forma2" action="" method="">
            <div id="content1" class="admin">
                <p class="left">Dodaj kategoriju</p>
                <input type="text"  class="right" id="category" autofocus="autofocus"
placeholder="Kategorija"/>
                <input type="button" value="DODAJ" class="btn" id="btn"/>
            </div>
        </form>
        <footer>
            <h5>
                <a href="{{route('genre-show')}}">Nazad na spisak kategorija</a>
            </h5>

        </footer>
    </div>
    <script src="{{asset("/")}}../js/jquery-3.3.1.min.js"
type="text/javascript"></script>
    <script src="{{asset("/")}}../js/adminInsertCategory.js"
type="text/javascript"></script>
```

dodaj-korisnika.blade.php

```
@extends('layouts.form')
@section('title','Dodaj korinsika')
@section('content')
<div id="wrapper" >
    <div id="box">
        <div id="errors">
            <noscript>
                <p class="false er">Molimo Vas omogućite JavuScript kako biste se
registrovali korisnika!</p>
            </noscript>
        </div>
        <p class="feedback ok" id="f200">Uspešno ste registrovali
korisnika!<br/></p>
        <p class="feedback er" id="f500">Trenutno ne možete registrovati
korisnika!</p>
        <div id="header">
            <h3>Registracija</h3>
        </div>
        <form id="forma2" action="" method="">
            <div id="content1" class="admin" >
                <p class="left">Ime</p>
                <input type="text"  class="right" id="first" autofocus="autofocus"
placeholder="Petar"/>
                <p class="left">Prezime</p>
                <input type="text" class="right" id="last" placeholder="Petrović"/>
                <p class="left">Email</p>
                <input type="text" class="right" id="email"
placeholder="petar@domen.com"/>
                <p class="left">Pol</p>
                <div id="pol" class="left">
                    <label for="m">Muški</label>
                    <input type="radio"  name="pol" class="radio" checked="checked"
id="m" value="m"/>
                    <label for="f">Ženski</label>
                    <input type="radio"  name="pol" class="radio"  id="f" value="z"/>
                </div>
                <p class="left">Grad</p>
                <select class="right select" name="city" id="city">
                    <option value="0">Izaberite...</option>
                    @foreach ($gradovi as $grad)
                    <option value="{{$grad->ID_grad}}">{{$grad-
>grad_ime}}</option>
                    @endforeach
                </select>
                <p class="left">Korisničko ime</p>
                <input type="text" class="right" id="username"
placeholder="korisnik1"/>
                <p class="left">Lozinka</p>
                <input type="password" class="right" id="password"
placeholder="Lozinka1#$%"/>
                <p class="left">Potvrdi lozinku</p>
                <input type="password" class="right" id="password1"
placeholder="Lozinka1#$%"/>
                <p class="left">Uloga</p>
                <select class="right select" name="roll" id="roll">
                    <option value="0">Izaberite...</option>
                    @foreach ($uloge as $uloga)
                        <option value="{{$uloga->ID_uloga}}">{{$uloga-
>uloga_ime}}</option>
                    @endforeach
                </select>
                <p class="left">Aktivnost</p>
                <select name="active" class="right select" id="active">
                    <option value="null">Izaberite...</option>
```

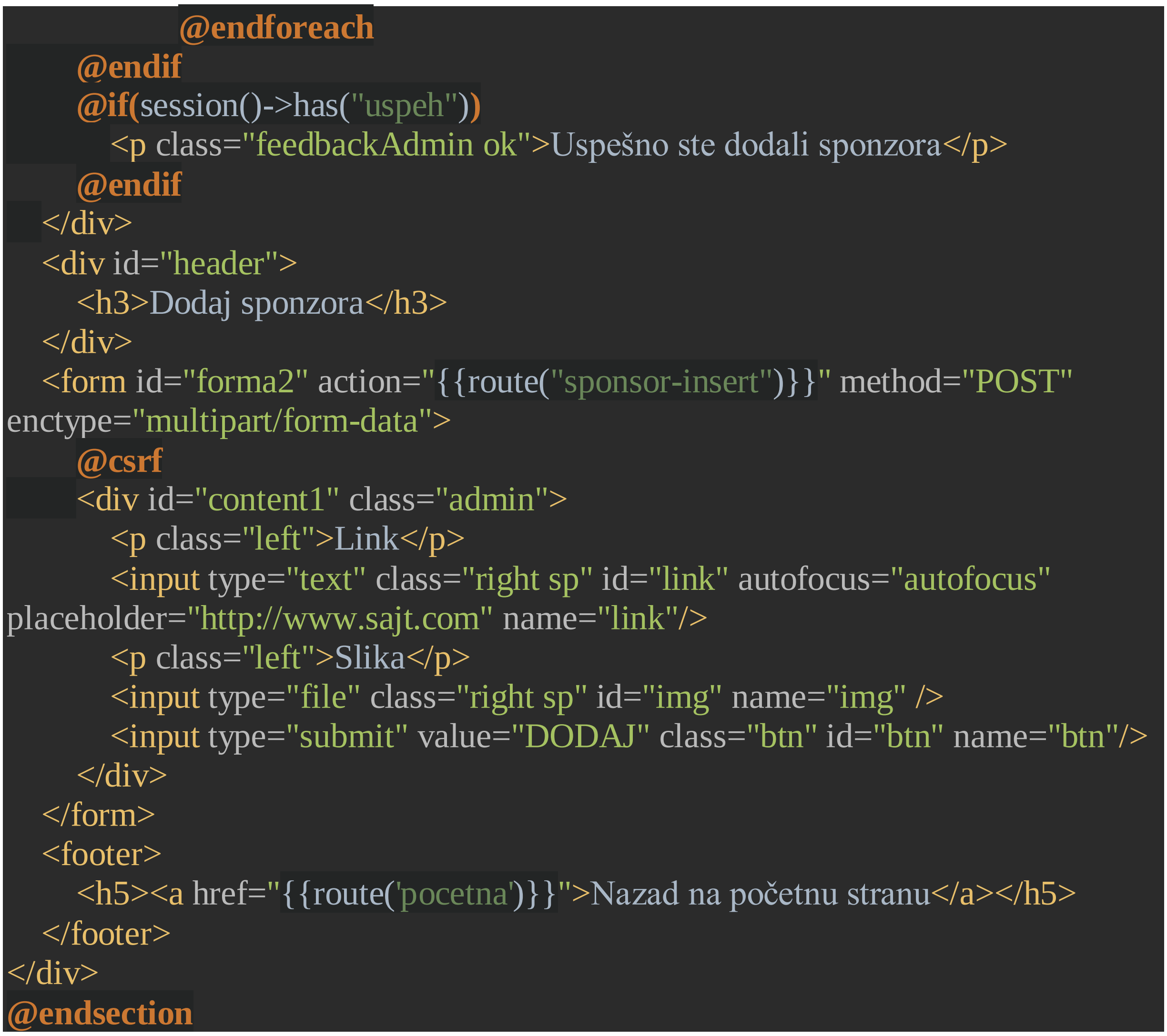
```
                    <option value="0">Neaktivan</option>
                    <option value="1">Aktivan</option>
                </select>
                <input type="button" value="REGISTRUJ" class="btn" id="btn"/>
            </div>
        </form>
        <footer>
            <h5>
                <a href="{{route('pocetna')}}">Nazad na početnu stranu</a>
            </h5>
        </footer>
        <script src="{{asset('/')}}js/jquery-3.3.1.min.js"
type="text/javascript"></script>
        <script src="{{asset('/')}}js/adminInsertUser.js"
type="text/javascript"></script>
@endsection
```

dodaj-seriju.blade.php

```
@extends('layouts.form')
@section('title','Dodaj seriju')
@section('content')
    <div id="wrapper">
        <div id="box">
            <div id="errors">
                @if(session()->has('error'))
                    <p class="feedbackAdmin er">{{session()->get('error')}}</p>
                @endif
                @if ($errors->any())
                    @foreach ($errors->all() as $error)
                        <p class="feedbackAdmin er">{{$error}}</p>
                    @endforeach
                @endif
                @if(session()->has("uspeh"))
                    <p class="feedbackAdmin ok">Uspešno ste dodali seriju</p>
                @endif
            </div>
            <div id="header">
                <h3>Dodaj seriju</h3>
            </div>
            <form id="forma2" action="{{route("series-insert")}}" method="POST"
enctype="multipart/form-data">
                @csrf
                <div id="admin-news" class="admin">
                    <p class="left">Izaberi kategoriju</p>
                    <select class="right select" id="category" name="category" >
                        <option value="0">Izaberite...</option>
                        @foreach ($kategorije as $kategorija)
                            <option  value="{{$kategorija->ID_link}}">{{$kategorija-
>naziv}}</option>
                        @endforeach
                    </select>
                    <p class="left">Izaberi glumce</p>
                    @foreach($glumci as $glumac)
                        <label  for="{{$glumac->ID_glumci}}">{{$glumac-
>glumci_ime}} {{$glumac->glumci_prezime}}</label>
                        <input  type="checkbox" id="{{$glumac->ID_glumci}}"
value="{{$glumac->ID_glumci}}" name="glumci[]" > <br/>
                    @endforeach
                    <p class="left">Ime serije</p>
                    <input type="text" class="right" name="name"  />
                    <p class="left">Godina</p>
                    <input type="text" class="right" name="year"  />
                    <p class="left">Tekst</p>
                    <textarea name="text" cols="30" rows="10"></textarea>
                    <p class="left">Slika</p>
                    <input class="right" type="file" name="img" >
                    <input type="submit" value="Dodaj" class="btn btn2" name="btn"/>
                </div>
            </form>
            <footer>
                <h5><a href="{{route('pocetna')}}">Nazad na početnu stranu</a></h5>
            </footer>
        </div>
@endsection
```

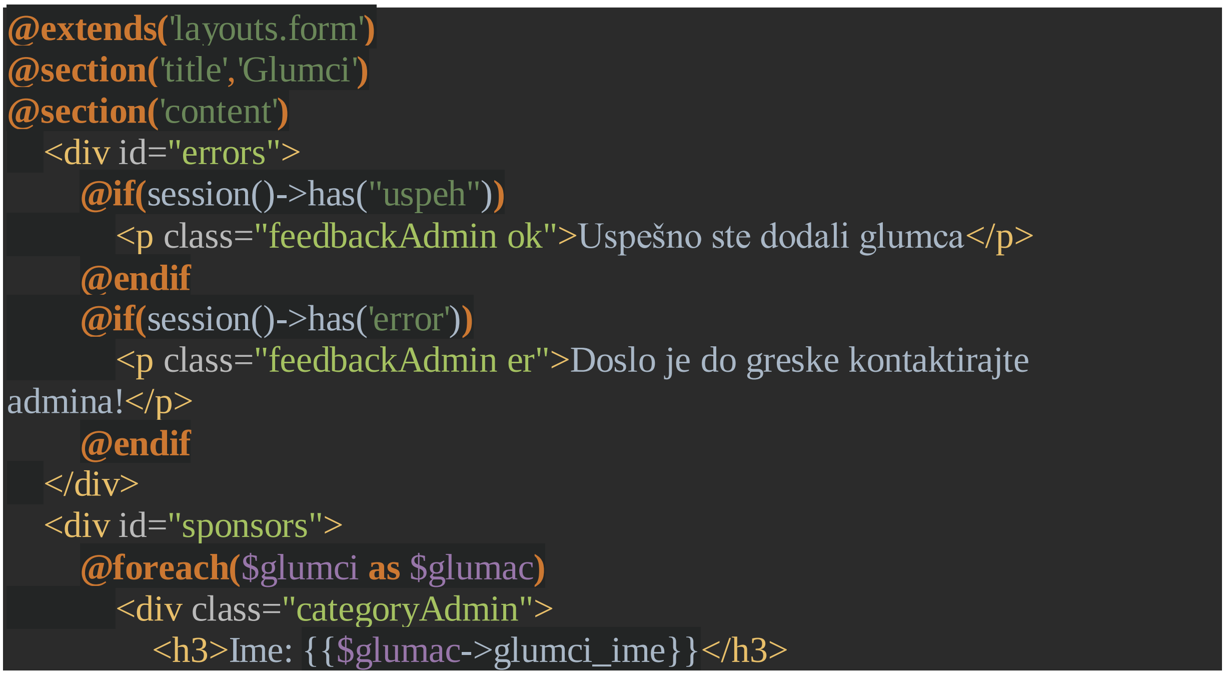
dodaj-sponzora.blade.php

```
@extends('layouts.form')
@section('title','Dodaj sponzora')
@section('content')
    <div id="wrapper">
<div id="box">
    <div id="errors">
        @if(session()->has('error'))
            <p class="feedbackAdmin er">{{session()->get('error')}}</p>
        @endif
        @if ($errors->any())
                    @foreach ($errors->all() as $error)
                        <p class="feedbackAdmin er">{{$error}}</p>
```



```
                    @endforeach
        @endif
        @if(session()->has("uspeh"))
            <p class="feedbackAdmin ok">Uspešno ste dodali sponzora</p>
        @endif
    </div>
    <div id="header">
        <h3>Dodaj sponzora</h3>
    </div>
    <form id="forma2" action="{{route("sponsor-insert")}}" method="POST"
enctype="multipart/form-data">
        @csrf
        <div id="content1" class="admin">
            <p class="left">Link</p>
            <input type="text" class="right sp" id="link" autofocus="autofocus"
placeholder="http://www.sajt.com" name="link"/>
            <p class="left">Slika</p>
            <input type="file" class="right sp" id="img" name="img" />
            <input type="submit" value="DODAJ" class="btn" id="btn" name="btn"/>
        </div>
    </form>
    <footer>
        <h5><a href="{{route('pocetna')}}">Nazad na početnu stranu</a></h5>
    </footer>
</div>
@endsection
```

glumci.blade.php



```
@extends('layouts.form')
@section('title','Glumci')
@section('content')
    <div id="errors">
        @if(session()->has("uspeh"))
            <p class="feedbackAdmin ok">Uspešno ste dodali glumca</p>
        @endif
        @if(session()->has('error'))
            <p class="feedbackAdmin er">Doslo je do greske kontaktirajte
admina!</p>
        @endif
    </div>
    <div id="sponsors">
        @foreach($glumci as $glumac)
            <div class="categoryAdmin">
                <h3>Ime: {{$glumac->glumci_ime}}</h3>
                <h3>Prezime: {{$glumac->glumci_prezime}}</h3>
                <button class="delSponsor" data-id='{{$glumac-
>ID_glumci}}'>Obriši</button>
                <a href="{{route('role-one-render',["id" => $glumac-
>ID_glumci])}}"><button class="editSponsor">Izmena</button></a>
            </div>
        @endforeach
    </div>
    <footer>
        <h5>
            <a href="{{route('pocetna')}}">Nazad na početnu stranu</a>
        </h5>
    </footer>
    <script type="text/javascript" src="{{asset('/')}}js/jquery-3.3.1.min.js"></script>
    <script src="{{asset("/")}}js/adminDeleteRole.js"
type="text/javascript"></script>
@endsection
```

izmena-glumca.blade.php

```
@extends('layouts.form')
@section('title','Izmena glumca')
<div id="wrapper">
    <div id="box">
        <div id="errors">
            <noscript>
                <p class="false er">Molimo Vas omogućite JavuScript kako biste
izmenili glumca!</p>
            </noscript>
        </div>
        <p class="feedback ok" id="f200">Uspešno ste izmenili glumca!<br/></p>
        <div id="header">
            <h3>Izmeni glumca</h3>
        </div>
        <form id="forma2" action="" method="">
            <div id="content1" class="admin">
                @foreach($podaci as $podatak)
                <input type="hidden" id="ID" value="{{$podatak->ID_glumci}}">
                <p class="left">Ime</p>
                <input type="text"  class="right" value="{{$podatak->glumci_ime}}"
id="ime" autofocus="autofocus" placeholder="Petar"/>
                <p class="left">Prezime</p>
                <input type="text"  class="right" id="prezime" value="{{$podatak-
>glumci_prezime}}" autofocus="autofocus" placeholder="Petrovic"/>
                <input type="button" value="IZMENI" class="btn" id="btn"/>
                @endforeach
            </div>
        </form>
        <footer>
            <h5>
                <a href="{{route('role-show')}}">Nazad na spisak glumaca</a>
            </h5>
        </footer>
    </div>
    <script src="{{asset("/")}}../js/jquery-3.3.1.min.js"
type="text/javascript"></script>
    <script src="{{asset("/")}}../js/adminUpdateRole.js"
type="text/javascript"></script>
```

izmena-kategorije.blade.php

```
@extends('layouts.form')
@section('title','Dodaj kategoriju')
<div id="wrapper" >
    <div id="box">
        <div id="errors">
            <noscript>
                <p class="false er">Molimo Vas omogućite JavuScript kako biste
izmenili kategoriju!</p>
            </noscript>
        </div>
        <p class="feedback ok" id="f200">Uspešno ste izmenili kategoriju!<br/></p>
        <p class="feedback er" id="f500">Trenutno ne možete izmeniti
kategoriju</p>
        <div id="header">
            <h3>Izmeni kategoriju</h3>
        </div>
        <form id="forma2" action="" method="">
            <div id="content1" class="admin">
                <p class="left">Izmeni kategoriju</p>
                @foreach($podaci as $podatak)
                    <input type="hidden" value="{{$podatak->ID_link}} " id='ID'>
                <input type="text"  class="right" id="category" value="{{$podatak-
>naziv}}" autofocus="autofocus" placeholder="Kategorija"/>
                <input type="button" value="DODAJ" class="btn" id="btn"/>
                @endforeach
            </div>
        </form>
        <footer>
            <h5>
```

```
                <a href="{{route('genre-show')}}">Nazad na spisak kategorija</a>
            </h5>
        </footer>
    </div>
    <script src="{{asset("/")}}../js/jquery-3.3.1.min.js"
type="text/javascript"></script>
    <script src="{{asset("/")}}../js/adminUpdateCategory.js"
type="text/javascript"></script>
```

izmena-serije.blade.php

```
@extends('layouts.form')
@section('title','Izmeni seriju')
@section('content')
    <div id="wrapper">
        <div id="box">
            <div id="errors">
                @if ($errors->any())
                    @foreach ($errors->all() as $error)
                        <p class="feedbackAdmin er">{{$error}}</p>
                    @endforeach
                @endif
            </div>
            <div id="header">
                <h3>Izmeni seriju</h3>
            </div>
            @foreach($serija as $s)
                <form id="forma2" action="{{route("series-update-info", ["id" => $s-
>ID_serija] )}}" method="POST" >
                    @method('PUT')
                    @csrf
                    <div id="edit-news" class="admin">
                        <p class="left">Naziv serije</p>
                        <input type="text" class="right sp"  autofocus="autofocus"
value="{{$s->serija_naziv}}"  name="name"/>
                        <p class="left">Godina</p>
                        <input type="text" class="right sp"  value="{{$s->godina}}"
name="year"/>
                        <p class="left">Tekst</p>
                        <textarea name="text" cols="30" rows="10">{{$s-
>serija_tekst}}</textarea>
                        <p class="left">Izaberi kategoriju</p>
                        <select class="right select" id="category" name="category">
                            <option  value="{{$s->ID_link}}">{{$s->naziv}}</option>
                            @foreach ($kategorije as $kategorija)
```

```
              <option  value="{{$kategorija->ID_link}}">{{$kategorija-
>naziv}}</option>
              @endforeach
            </select>
            @foreach($svi_glumci as $glumac)
              <label for="{{$glumac->ID_glumci}}">{{$glumac-
>glumci_ime}} {{$glumac->glumci_prezime}}</label>
              <input type="checkbox" id="{{$glumac->ID_glumci}}"
value="{{$glumac->ID_glumci}}" name="glumci[]"
                 @foreach($glumci as $g)
                  @if($g->ID_glumci == $glumac->ID_glumci)
                  checked="checked"
                  @endif
                  @endforeach>
              <br/>

            @endforeach
            <p class="left">Status</p>
            <select name="status" class="right select" id="a">
              <option value="null">Izaberite...</option>
              @if($s->obrisan == 0)
                <option selected="selected" value="0">Nije obrisan</option>
                <option  value="1">Obrisan</option>
              @else
                <option  value="0">Nije obrisan</option>
                <option selected="selected" value="1">Obrisan</option>
              @endif
            </select>
            <input type="submit" value="IZMENI" class="btn btn2" id="btn2"
/>
      @endforeach
          </div>
          <footer>
            <h5><a href="{{route('pocetna')}}">Nazad na početnu
stranu</a></h5>
          </footer>
        </form>
    </div>
    <div id="box">
      <div id="errors">
        @if($errors->any())
          @foreach ($errors->all() as $error)
            <p class="feedbackAdmin er">{{$error}}</p>
          @endforeach
        @endif
        @if(session()->has('uspeh'))
```

```
                        <p class="feedbackAdmin ok">Uspesno ste izmenili sliku</p>
                @endif
            </div>
            <div id="header">
                <h3>Izmeni sliku</h3>
            </div>
            @foreach($serija as $s)
                <form id="forma3" action="{{route("series-update-picture", ["id" => $s->ID_serija] )}}" method="POST" enctype="multipart/form-data">
                    @csrf
                    @method('PUT')
                    <div id="content1" class="admin">
                        <p class="left">Slika</p>
                        <input type="file" class="right sp" id="img" name="img" />
                        @endforeach
                        <input type="submit" value="IZMENI" class="btn" id="btn" name="btn"/>
                    </div>
                </form>
                <footer>
                    <h5><a href="{{route('pocetna')}}">Nazad na početnu stranu</a></h5>
                </footer>
        </div>
    </div>
@endsection
```

izmena-sponzora.blade.php

```
@extends('layouts.form')
@section('title','Izmeni sponzora')
@section('content')
    <div id="wrapper">
    <div id="box">
        <div id="errors">
            @if ($errors->any())
                @foreach ($errors->all() as $error)
                    <p class="feedbackAdmin er">{{$error}}</p>
                @endforeach
            @endif
        </div>
        <div id="header">
            <h3>Izmeni link</h3>
        </div>
        @foreach($podaci as $podatak)
        <form id="forma2" action="{{route("sponsor-update-link", ["id" =>
$podatak->ID_sponzor] )}}" method="POST" enctype="multipart/form-data">
            @method('PUT')
            @csrf

            <div id="content1" class="admin">
                <p class="left">Link</p>
                <input type="text" class="right sp" id="link" autofocus="autofocus"
value="{{$podatak->link}}" placeholder="http://www.sajt.com" name="link"/>
        @endforeach
                <input type="submit" value="IZMENI" class="btn" id="btn"
name="btn"/>
            </div>
            <footer>
                <h5><a href="{{route('pocetna')}}">Nazad na početnu stranu</a></h5>
            </footer>
        </form>
        </div>
            <div id="box">
                <div id="errors">
                    @if ($errors->any())
                        @foreach ($errors->all() as $error)
                            <p class="feedbackAdmin er">{{$error}}</p>
                        @endforeach
                    @endif
                </div>
                <div id="header">
                    <h3>Izmeni sliku</h3>
                </div>
                @foreach($podaci as $podatak)
                    <form id="forma3" action="{{route("sponsor-update-picture", ["id"
=> $podatak->ID_sponzor] )}}" method="POST" enctype="multipart/form-data">
                        @csrf
                        @method('PUT')
                        <div id="content1" class="admin">
                            <p class="left">Slika</p>
                            <input type="file" class="right sp" id="img" name="img" />
                            @endforeach
                            <input type="submit" value="IZMENI" class="btn" id="btn"
name="btn"/>
                        </div>
                    </form>
                    <footer>
                        <h5><a href="{{route('pocetna')}}">Nazad na početnu
stranu</a></h5>
                    </footer>
    </div>
    </div>
@endsection
```

izmena-korisnika.blade.php

```
@extends('layouts.form')
@section('title', 'Spisak korisnika')
@section('content')
<div id="wrapper">
    <div id="box">
        <div id="errors">
            <noscript>
                <p class="false er">Molimo Vas omogućite JavuScript kako biste se izmenili korisnika!</p>
            </noscript>
        </div>
        <p class="feedback ok" id="f200">Uspešno ste izmenili korisnika!<br/></p>
        <p class="feedback er" id="f500">Trenutno ne možete izmeniti korisnika!</p>
        <div id="header">
            <h3>Izmena</h3>
        </div>
        @foreach($podaci as $podatak)
        <form id="forma2" action="" method="">
            <div id="content1" class="admin">
                <input type="hidden" value="{{$podatak->ID_korisnik}}" id="ID">
                <p class="left">Ime</p>
                <input type="text"  class="right" id="first" autofocus="autofocus" value="{{$podatak->ime}}"/>
                <p class="left">Prezime</p>
                <input type="text" class="right" id="last" value="{{$podatak->prezime}}"/>
                <p class="left">Email</p>
                <input type="text" class="right" id="email" value="{{$podatak->email}}"/>
                <p class="left">Pol</p>
                <div id="pol" class="left">
                    <label for="m">Muški</label>
                    <input type="radio" name="pol" class="radio" @if($podatak->pol=="m") checked="checked" @endif id="m" value="m"/>
                    <label for="f">Ženski</label>
                    <input type="radio"  name="pol" class="radio" @if($podatak->pol=="z") checked="checked" @endif id="f" value="z"/>
                </div>
                <p class="left">Grad</p>
                <select class="right select" name="city" id="city">
                    <option value="0">Izaberite...</option>
                    <option selected="selected" value="{{$podatak->ID_grad}}">{{$podatak->grad_ime}}</option>
                    @foreach ($gradovi as $grad)
                        <option value="{{$grad->ID_grad}}">{{$grad->grad_ime}}</option>
                    @endforeach
                </select>
                <p class="left">Korisničko ime</p>
                <input type="text" class="right" id="username" value="{{$podatak->username}}"/>
                <p class="left">Uloga</p>
                <select class="right select" name="roll" id="roll">
                    <option value="0">Izaberite...</option>
                    <option selected="selected" value="{{$podatak->ID_uloga}}">{{$podatak->uloga_ime}}</option>
                    @foreach ($uloge as $uloga)
                    <option value="{{$uloga->ID_uloga}}">{{$uloga->uloga_ime}}</option>
                    @endforeach
                </select>
                <p class="left">Aktivnost</p>
                <select name="active" class="right select" id="active">
                    <option value="null">Izaberite...</option>
                    @if($podatak->aktivan === 0)
                    <option selected="selected" value="0">Neaktivan</option>
                    <option  value="1">Aktivan</option>
                    @else
                    <option  value="0">Neaktivan</option>
                    <option selected="selected" value="1">Aktivan</option>
                    @endif
                </select>
                <p class="left">Status</p>
                <select name="status" class="right select" id="a">
                    <option value="null">Izaberite...</option>
                    @if($podatak->obrisan == 0)
                    <option selected="selected" value="0">Nije obrisan</option>
                    <option  value="1">Obrisan</option>
                    @else
                    <option  value="0">Nije obrisan</option>
                    <option selected="selected" value="1">Obrisan</option>
                    @endif
                </select>
                <input type="button" value="Izmeni" class="btn" id="btn"/>
                @endforeach
            </div>
        </form>
        <footer>
            <h5>
                <a href="{{route('user-show')}}">Nazad na spisak korisnika</a>
            </h5>
        </footer>
    </div>
    <script src="{{asset('/')}}../js/jquery-3.3.1.min.js"
type="text/javascript"></script>
    <script src="{{asset('/')}}../js/adminUpdateUser.js"
type="text/javascript"></script>
@endsection
```

izvestaj.blade.php

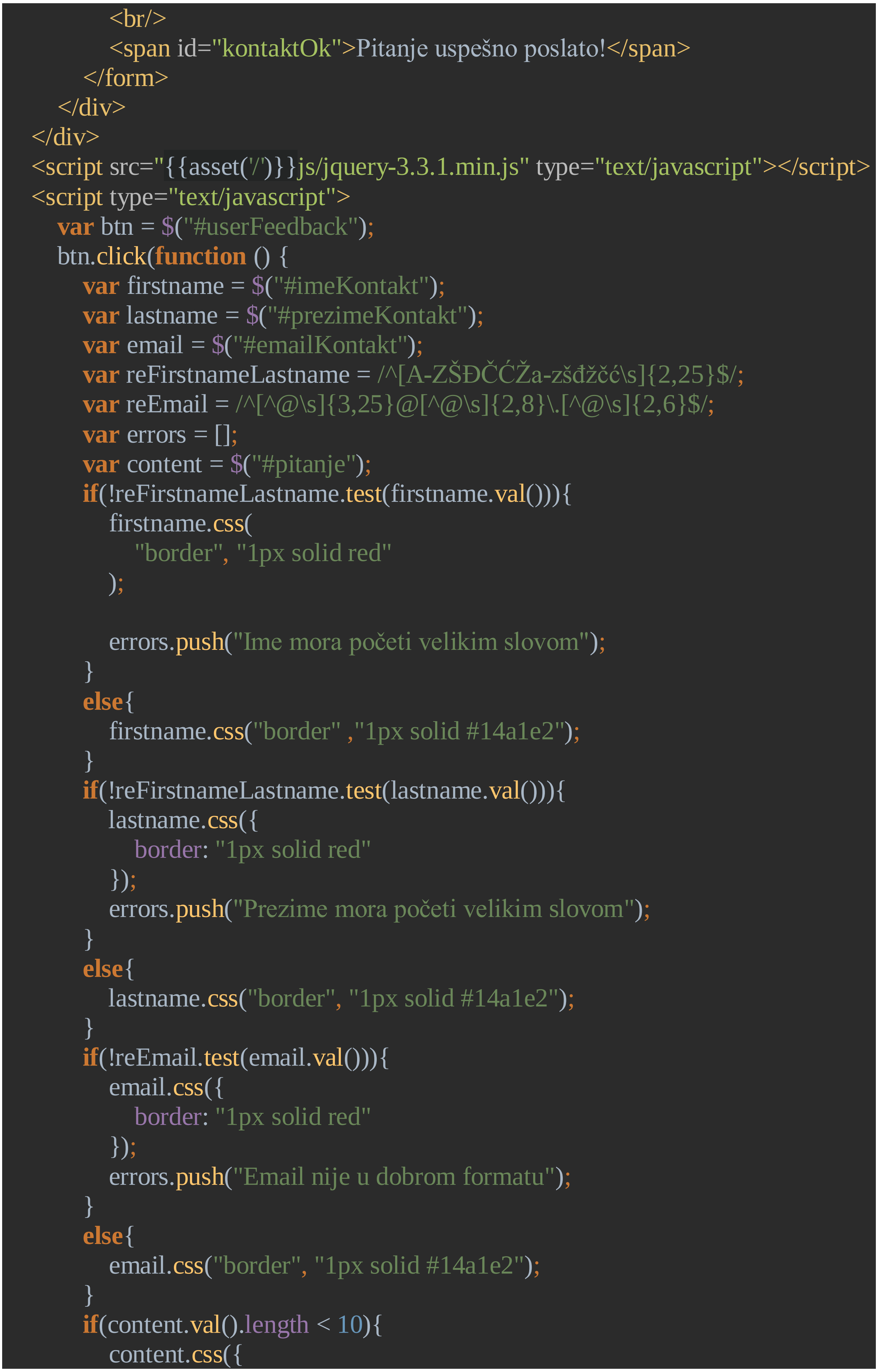
```
@extends('layouts.form')
@section('title', 'Izvestaj aktivnosti')
<table>
    <tr class="row root">
        <th class="cell">Username</th>
        <th class="cell">Datum</th>
        <th class="cell">Aktivnost</th>
    </tr>
    @foreach($data as $d)
    <tr class="row">
        <td class="cell">{{$d->username}}</td>
        <td class="cell">{{$d->datum}}</td>
        <td class="cell">{{$d->aktivnost}}</td>
    </tr>
    @endforeach

</table>
<footer>
    <h5><a  href="{{route('pocetna')}}">Nazad na početnu stranu</a></h5>
</footer>
```

kategorije.blade.php

```
@extends('layouts.form')
@section('title','Kategorije')
@section('content')
    <div id="errors">
        @if(session()->has("uspeh"))
            <p class="feedbackAdmin ok">Uspešno ste dodali kategoriju</p>
        @endif
        @if(session()->has('error'))
            <p class="feedbackAdmin er">Doslo je do greske kontaktirajte
admina!</p>
        @endif
    </div>
    <div id="sponsors">
        @foreach($kategorije as $kategorija)
            <div class="categoryAdmin">
                <h3>{{$kategorija->naziv}}</h3>
                <button class="delSponsor" data-id='{{$kategorija-
>ID_link}}'>Obriši</button>
                <a href="{{route('genre-one-render',["id" => $kategorija-
>ID_link])}}"><button class="editSponsor" >Izmena</button></a>
            </div>
        @endforeach

    </div>
    <footer>
        <h5>
            <a href="{{route('pocetna')}}">Nazad na početnu stranu</a>
        </h5>
    </footer>
    <script type="text/javascript" src="{{asset('/')}}js/jquery-3.3.1.min.js"></script>
    <script src="{{asset("/")}}js/adminDeleteCategory.js"
type="text/javascript"></script>
@endsection
```

kontakt.blade.php

```
@extends('layouts.master')
@section('title','Kontakt')
@section('content')
    <div id="articles">
        <div id="kontakt_forma">

            <form id="obrazac1" >
                <input type="text" id="imeKontakt" autofocus="autofocus"
placeholder="Ime"/>
                <input type="text" id="prezimeKontakt" placeholder="Prezime"/>
                <input type="text"  id="emailKontakt"  placeholder="Email"/>
                <textarea id="pitanje"   placeholder="Vaše pitanje"></textarea>
                <input type="button" value="Pošalji" class="btn" name="contact"
id="userFeedback"/>
                <br/>
```



```
            <br/>
            <span id="kontaktOk">Pitanje uspešno poslato!</span>
        </form>
    </div>
</div>
<script src="{{asset('/')}}js/jquery-3.3.1.min.js" type="text/javascript"></script>
<script type="text/javascript">
    var btn = $("#userFeedback");
    btn.click(function () {
        var firstname = $("#imeKontakt");
        var lastname = $("#prezimeKontakt");
        var email = $("#emailKontakt");
        var reFirstnameLastname = /^[A-ZŠĐČĆŽa-zšđžčć\s]{2,25}$/;
        var reEmail = /^[^@\s]{3,25}@[^@\s]{2,8}\.[^@\s]{2,6}$/;
        var errors = [];
        var content = $("#pitanje");
        if(!reFirstnameLastname.test(firstname.val())){
            firstname.css(
                "border", "1px solid red"
            );

            errors.push("Ime mora početi velikim slovom");
        }
        else{
            firstname.css("border" ,"1px solid #14a1e2");
        }
        if(!reFirstnameLastname.test(lastname.val())){
            lastname.css({
                border: "1px solid red"
            });
            errors.push("Prezime mora početi velikim slovom");
        }
        else{
            lastname.css("border", "1px solid #14a1e2");
        }
        if(!reEmail.test(email.val())){
            email.css({
                border: "1px solid red"
            });
            errors.push("Email nije u dobrom formatu");
        }
        else{
            email.css("border", "1px solid #14a1e2");
        }
        if(content.val().length < 10){
            content.css({
                    border: "1px solid red"
                });
                errors.push("Sadrzaj je prekratak");
            }
            else{
                content.css("border","1px solid #14a1e2");
            }
            if(errors.length > 0)
                console.log("Greska");
            else {
                $.ajaxSetup({
                    headers:{
                        'X-CSRF-TOKEN' : $('meta[name="_token"]').attr('content')
                    }
                });
                $.ajax({
                    url: "/feedback",
                    method: "POST",
                    data: {
                        ime: firstname.val(),
                        prezime: lastname.val(),
                        email: email.val(),
                        content: content.val()
                    },
                    success: function(data, xhr) {
                        $("#kontaktOk").css("display", "block");
                    },
                    error: function(xhr, status, error){
                        alert("Izvinjavamo se, trenutno imamo tehničkih problema");
                    }
                });
            }
        });
    </script>
@endsection
```

korisnici.blade.php

```
@extends('layouts.form')
@section('title', 'Spisak korisnika')
@section('content')
    <div id="users">
        @foreach ($korisnici as $korisnik)
        <div class="item">
            <p class="key" >Ime</p>
            <p class="value">{{$korisnik->ime}}</p>
            <p class="key">Prezime</p>
            <p class="value">{{$korisnik->prezime}}</p>
            <p class="key">Korisničko ime</p>
            <p class="value">{{$korisnik->username}}</p>
            <p class="key">Email</p>
            <p class="value">{{$korisnik->email}}</p>
            <p class="key">Aktivnost</p>
            <p class="value">{{$korisnik->aktivan}}</p>
            <p class="key">Status</p>
            <p class="value">
                @if($korisnik->obrisan==0) {{"Nije obrisan"}}
                @else  {{"Obrisan"}}</p>
                @endif
            @if($korisnik->datum_izmene=="/")
            <p class="key">Datum registracije</p>
            <p class="value">{{$korisnik->datum_registracije}}</p>
            @else
            <p class="key">Datum izmene</p>
            <p class="value">{{$korisnik->datum_izmene}}</p>
            @endif
            <a href="{{route('user-one-render',["id" => $korisnik-
>ID_korisnik])}}"><button class="edit">Izmeni</button></a>
            <button @if($korisnik->obrisan==1) disabled="disabled" @endif
class="del" data-id="{{$korisnik->ID_korisnik}}">Obriši</button>
        </div>
        @endforeach
    </div>
    <footer>
        <h5><a  href="{{route('pocetna')}}">Nazad na početnu stranu</a></h5>
    </footer>
    <script src="{{asset("/")}}js/jquery-3.3.1.min.js"
type="text/javascript"></script>
    <script src="{{asset("/")}}js/adminDeleteUser.js"
type="text/javascript"></script>
@endsection
```

pocetna.blade.php

```
@extends('layouts.master')
@section('title','Pocetna')
@section('content')
    <div id="articles">
        @foreach($serije as $serija)
            <div class="item">
            <div class="img">
               <img src="{{asset("/").$serija->slika }}" alt="{{ $serija->serija_naziv
}}" title="{{ $serija->serija_naziv }}" width="200" height="150">
            </div>
            <div class="right-container">
               <div class="date-com">
                  @if(session()->has('user'))
                     <i data-user="{{session()->get('user')->ID_korisnik}}"
                       data-series="{{ $serija->ID_serija }}"
                       class="left vote-series-plus fa fa-plus"> {{ $serija->serija_like
}} </i>
                     <i data-user="{{session()->get('user')->ID_korisnik}}"
                       data-series="{{ $serija->ID_serija }}"
                       class="left vote-series-minus fa fa-minus"> {{$serija-
>serija_dislike}} </i>
                  @else
                     <a href="{{route('renderLogin')}}"><i class="left fa fa-
plus"></i><i  class="left fa fa-minus"></i></a>
                  @endif
                  @if(session()->has('user'))
                     <h6 class="right" data-idseries="{{ $serija->ID_serija}}"
                        data-user="{{ session()->get('user')->ID_korisnik}}"
                        data-series="{{ $serija->serija_naziv}}">Komentari</h6>
                  @else
                     <h6 class="right" data-idseries="{{ $serija->ID_serija}}"
                        data-series="{{ $serija->serija_naziv}}">Komentari</h6>
                  @endif

               </div>
               <div class="details" data-idseries="{{$serija->ID_serija}}">
                  <h3 class="heading">{{ $serija->serija_naziv }}</h3>
                  <p class="text" >{{ substr($serija->serija_tekst,0,150).'...' }}</p>
               </div>
            </div>

         </div>
      @endforeach
   </div>
@endsection
```

prijava.blade.php

```
@extends('layouts.form')
@section('title','Prijava')
@section('content')
```

```
    <div id="wrapper">
    <div id="box">
        <div id="errors">
            <noscript>
                <p class="false er">Molimo Vas omogućite JavuScript kako biste se prijavili!</p>
            </noscript>
        </div>
        <p class="feedback ok" id="f200">Uspešno ste se prijavili!</p>
        <p class="feedback er" id="f400">Uneti podaci nisu ispravni!</p>
        <p class="feedback er" id="f500">Trenutno se ne možete prijaviti!</p>
        <div id="header">
            <h3>Prijava</h3>
        </div>
        <form id="forma1" action="" method="">
            <div id="content">
                <p class="left">Korisničko ime</p>
                <input type="text" class="right" id="username" autofocus="autofocus" placeholder="korisnik1"/>
                <p class="left">Lozinka</p>
                <input type="password" class="right" id="password" placeholder="Lozinka1#$%"/>
                <input type="button" value="PRIJAVI SE" class="btn" id="btn" />
            </div>
        </form>
        <footer>
            <h5>
                Niste registrovani? <a href="{{route('renderRegistration')}}">Registrujte se</a> <br/>
                <a href="{{route('pocetna')}}">Nazad na početnu stranu</a>
            </h5>
        </footer>
    </div>
    <script src="{{asset('/')}}js/jquery-3.3.1.min.js" type="text/javascript"></script>
    <script src="{{asset('/')}}js/login.js" type="text/javascript"></script>
    @endsection
```

registracija.blade.php

```
@extends('layouts.form')
@section('title', 'Registracija')
@section('content')
    <div id="wrapper">
    <div id="box">
        <div id="errors">
```

```
            <noscript>
                <p class="false er">Molimo Vas omogućite JavuScript kako biste se registrovali!</p>
            </noscript>
        </div>
        <p class="feedback ok" id="f200">Uspešno ste se registrovali!<br/>Poslat Vam je email sa aktivacionim linkom!</p>
        <p class="feedback er" id="f500">Trenutno se ne možete registrovati!</p>
        <div id="header">
            <h3>Registracija</h3>
        </div>
        <form>
            <div id="content1" >
                <p class="left">Ime</p>
                <input type="text"  class="right" id="first" autofocus="autofocus" placeholder="Petar"/>
                <p class="left">Prezime</p>
                <input type="text" class="right" id="last" placeholder="Petrović"/>
                <p class="left">Email</p>
                <input type="text" class="right" id="email" placeholder="petar@domen.com"/>
                <p class="left">Pol</p>
                <div id="pol" class="left">
                    <label for="m">Muški</label>
                    <input type="radio"  name="pol" class="radio" checked="checked" id="m" value="m"/>
                    <label for="f">Ženski</label>
                    <input type="radio"  name="pol" class="radio"  id="f" value="z"/>
                </div>
                <p class="left">Grad</p>
                <select class="right select" name="city" id="city">
                    <option value="0">Izaberite...</option>
                    @foreach ($cities as $city)
                    <option value="{{$city->ID_grad}}">{{$city->grad_ime}}</option>
                    @endforeach
                </select>
                <p class="left">Korisničko ime</p>
                <input type="text" class="right" id="username" placeholder="korisnik1"/>
                <p class="left">Lozinka</p>
                <input type="password" class="right" id="password" placeholder="Lozinka1#$%"/>
                <p class="left">Potvrdi lozinku</p>
                <input type="password" class="right" id="password1" placeholder="Lozinka1#$%"/>
                <input type="button" value="REGISTRUJ SE" class="btn" id="btn"/>
            </div>
        </form>
        <footer>
            <h5>
                Imate nalog?  <a href="{{route("renderLogin")}}">Prijavite se</a>
<br/>
                <a href="{{route('pocetna')}}">Nazad na početnu stranu</a>
            </h5 >
        </footer>
    </div>
    </div>
    <script src="{{asset('/')}}js/jquery-3.3.1.min.js" type="text/javascript"></script>
    <script src="{{asset('/')}}js/registration.js" type="text/javascript"></script>
@endsection
```

serije.blade.php

```
@extends('layouts.master')
@section('content')
<div id="articles">
    @foreach($serije as $serija)
    <div class="item">
        <div class="img">
            <img src="{{asset("/").$serija->slika }}" alt="{{ $serija->serija_naziv }}"
title="{{ $serija->serija_naziv }}" width="200" height="150">
        </div>
        <div class="right-container">
            <div class="date-com">
                @if(session()->has('user'))
                    <i data-user="{{session()->get('user')->ID_korisnik}}"
                       data-idseries="{{ $serija->ID_serija }}"
                       class="left vote-series-plus fa fa-plus"> {{ $serija->serija_like }}
</i>
                    <i data-user="{{session()->get('user')->ID_korisnik}}"
                       data-idseries="{{ $serija->ID_serija }}"
                        class="left vote-series-minus fa fa-minus"> {{$serija-
>serija_dislike}} </i>
                @else
                    <a href="{{route('renderLogin')}}"><i class="left fa fa-plus"></i><i
class="left fa fa-minus"></i></a>
                @endif
                @if(session()->has('user'))
                     <h6 class="right" data-idseries="{{ $serija->ID_serija}}"
                         data-user="{{ session()->get('user')->ID_korisnik}}"
                         data-series="{{ $serija->serija_naziv}}">Komentari</h6>
                @else
                        <h6 class="right" data-idseries="{{ $serija->ID_serija}}"
                            data-series="{{ $serija->serija_naziv}}">Komentari</h6>
                @endif

            </div>
            <div class="details" data-idseries="{{$serija->ID_serija}}">
                <h3 class="heading">{{ $serija->serija_naziv }}</h3>
                <p class="text" >{{ substr($serija->serija_tekst,0,150).'...' }}</p>
            </div>
        </div>
        @section('title', $serija->naziv_kategorije)
    </div>
    @endforeach
</div>
@endsection
```

serije-spisak.blade.php

```
@extends('layouts.form')
@section('title','Serije')
@section('content')
    <div id="admin-news">
        <div id="errors">
            @if(session()->has('error'))
                <p class="feedbackAdmin er">{{session()->get('error')}}</p>
            @endif
            @if(session()->has("uspeh"))
                <p class="feedbackAdmin ok">{{session()->get('uspeh')}}</p>
            @endif
        </div>
        @foreach($serije as $serija)
        <div class="series">
            <div class="img">
                <img src="{{ asset('/').$serija->slika }}" alt="{{ $serija->serija_naziv
}}" title="{{ $serija->serija_naziv }}" width="200" height="150">
            </div>
            <div class="text padd">
                <span class="heading">
            <h6>Godina: {{ $serija->godina }}</h6>
            <h3>{{ $serija->serija_naziv }}</h3>
                    <h4>Status : @if($serija->obrisan==1) {{"Obrisano"}} @else {{"Nije
obrisano"}} @endif</h4>
        </span>
                <div class="details">{{ substr($serija->serija_tekst,0,150).'...' }} <br/>
```

```
                    <a href="{{route('series-one-render',["id" => $serija->ID_serija, "cat"
=> $serija->ID_link])}}"><button class="edit s">Izmeni</button></a>
                    <button class="del s" data-id="{{$serija-
>ID_serija}}">Obriši</button>
                </div>
            </div>
        </div>
        @endforeach
    </div>
    <footer>
        <h5><a  href="{{route('pocetna')}}">Nazad na početnu stranu</a></h5>
    </footer>
    <script src="{{asset("/")}}js/jquery-3.3.1.min.js"
type="text/javascript"></script>
    <script src="{{asset("/")}}js/adminDeleteSeries.js"
type="text/javascript"></script>
    @endsection
```

sponzori.blade.php

```
@extends('layouts.form')
@section('content')
    <div id="errors">
        @if(session()->has("uspeh"))
            <p class="feedbackAdmin ok">Uspešno ste izmenili sponzora</p>
        @endif
        @if(session()->has('error'))
                <p class="feedbackAdmin er">Doslo je do greske kontaktirajte
admina!</p>
        @endif
    </div>
    <div id="sponsors">
        @foreach($sponzori as $item)
        <div class="sponsorAdmin">
            <button class="delSponsor" data-id='{{$item-
>ID_sponzor}}'>Obriši</button>
            <a href="{{route('sponsor-one-render',["id" => $item-
>ID_sponzor])}}"><button class="editSponsor" >Izmena</button></a>
            <a target="_blank" href="{{ $item->link }}">
                <img src="{{asset("/").$item->slika }}" alt="{{ substr($item->link,7)
}}" title="{{ substr($item->link,7) }}" width="300" height="150">
            </a>
        </div>
        @endforeach
    </div>
    <script type="text/javascript" src="{{asset('/')}}js/jquery-3.3.1.min.js"></script>
    <script type="text/javascript">
        $('.delSponsor').click(function () {
            var id = $(this).data('id');
                $.ajaxSetup({
                    headers: {
                        "X-CSRF-TOKEN": $("meta[name='_token']").attr("content")
                    }
                });
            console.log(id);
            $.ajax({
                url:"/admin/sponzori/brisanje/"+id,
                method:"DELETE",
                success:function (data) {
                     alert("Uspešno ste obrisali sponzora!");
                     location.reload();
                },
                error:function (x,s,e) {
                    switch (x.status) {
                        case 404:
                            window.location.href="greska.php";
                            break;
                        case 409:
                            alert("Izvinjavamo se zbog tehničkih problema");
                            break;
                    }
                }
            })
        });
    </script>
    </body>
    </html>
@endsection
```

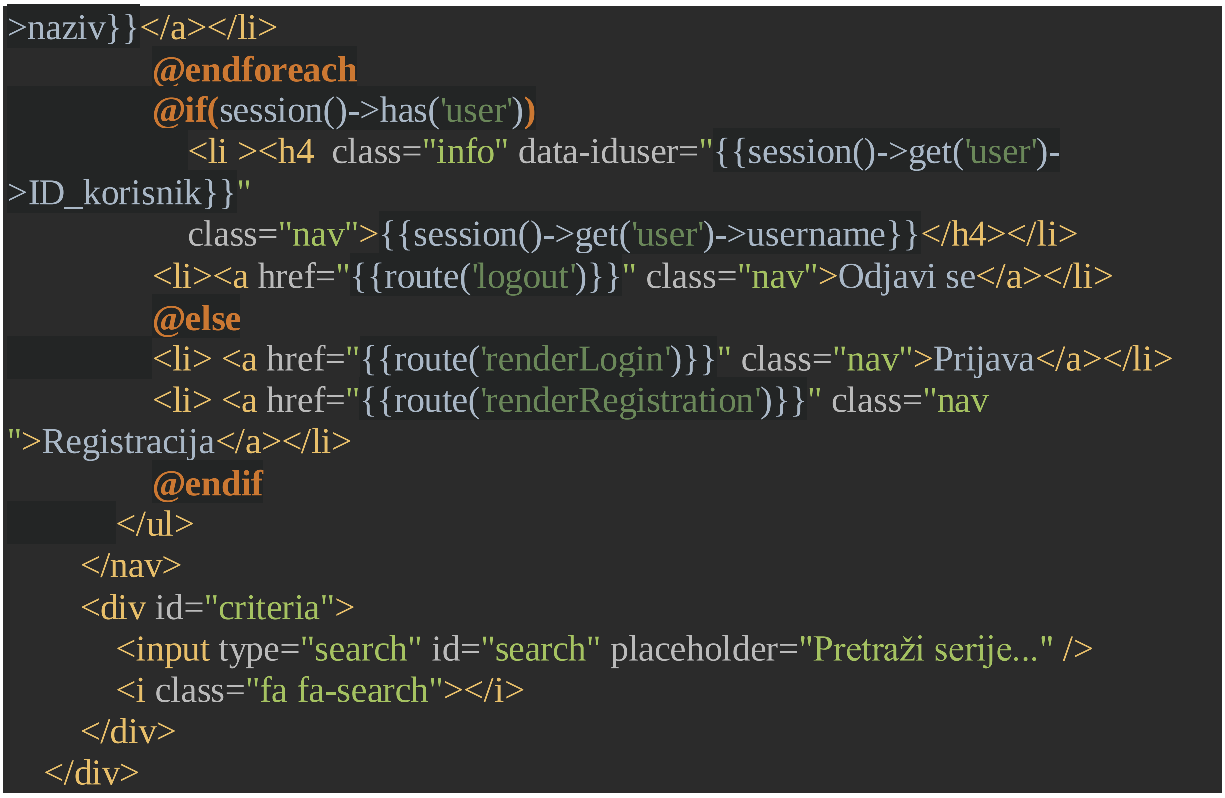
------------------------resources/views/partials-------------------------------

footer.blade.php

```
<footer>
    <a href="{{asset('/')}}dokumentacija.pdf">Dokumentacija</a>
    <h6>Copyright Dušan Krsmanović {{date("Y")}} &copy;</h6>
    @if(session()->has('user') && session()->get('user')->ID_uloga == 1)
        <a href="{{route("sponsor-show")}}">Spisak sponzora</a>
        <a href="{{route('user-show')}}">Spisak korisnika</a>
        <a href="{{route('genre-show')}}">Spisak žanrova</a>
        <a href="{{route("role-show")}}">Spisak glumaca</a>
        <a href="{{route('series-show')}}">Spisak serija</a>
        <a href="{{route('user-form')}}">Dodaj korisnika</a>
        <a href="{{route("genre-form")}}">Dodaj žanr</a>
        <a href="{{route('sponsor-form')}}">Dodaj sponzora</a>
        <a href="{{route('series-form')}}">Dodaj seriju</a>
        <a href="{{route('role-form')}}">Dodaj glumce</a>
        <a href="{{route('activity')}}">Izveštaj</a>
    @endif
</footer>
</div>
<script type="text/javascript" src="{{asset('/')}}js/jquery-3.3.1.min.js"></script>
<script type="text/javascript" src="{{asset('/')}}js/main.js"></script>
</body>
</html>
```

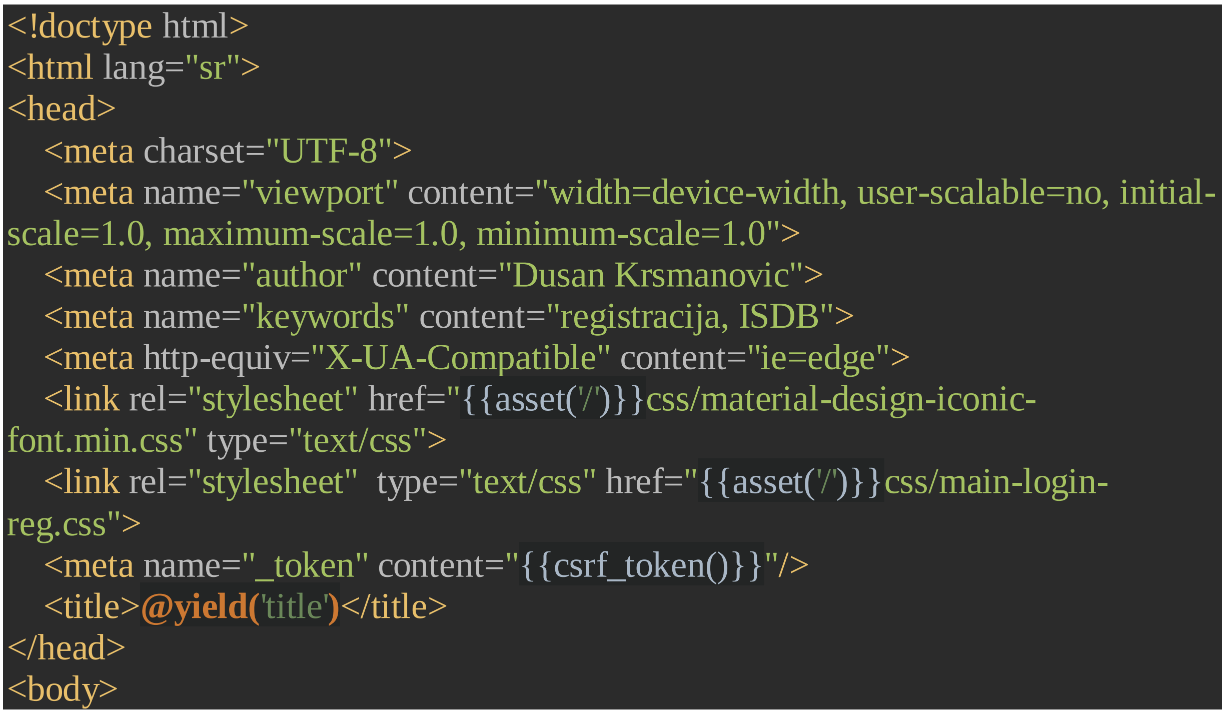
header.blade.php

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
          content="width=device-width, user-scalable=no, initial-scale=1.0,
maximum-scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="stylesheet"
href="https://use.fontawesome.com/releases/v5.6.3/css/all.css" integrity="sha384-
UHRtZLI+pbxtHCWp1t77Bi1L4ZtiqrqD80Kn4Z8NTSRyMA2Fd33n5dQ8lWUE0
0s/" crossorigin="anonymous">
    <link rel="stylesheet" href="{{asset('/')}}css/main.css" type="text/css">
    <meta name="_token" content="{{csrf_token()}}"/>
    <title>ISDB @yield('title')</title>
</head>
<body>
<div id="wrapper">
    <div id="header">
        <div id="logo">
            <a href=""><img src="{{asset("/")}}images/logo.png" alt="logo"
title="logo" height="98" ></a>
        </div>
        <nav>
            <ul>
                @foreach($linkovi as $link)
                <li><a href="{{$link->putanja}}" class="nav">{{$link-
```



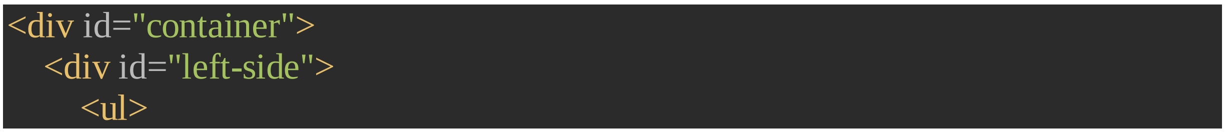
```
>naziv}}</a></li>
                @endforeach
                @if(session()->has('user'))
                    <li ><h4  class="info" data-iduser="{{session()->get('user')-
>ID_korisnik}}"
                    class="nav">{{session()->get('user')->username}}</h4></li>
                <li><a href="{{route('logout')}}" class="nav">Odjavi se</a></li>
                @else
                <li> <a href="{{route('renderLogin')}}" class="nav">Prijava</a></li>
                <li> <a href="{{route('renderRegistration')}}" class="nav
">Registracija</a></li>
                @endif
            </ul>
        </nav>
        <div id="criteria">
            <input type="search" id="search" placeholder="Pretraži serije..." />
            <i class="fa fa-search"></i>
        </div>
    </div>
```

headForm.blade.php



```
<!doctype html>
<html lang="sr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, user-scalable=no, initial-
scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
    <meta name="author" content="Dusan Krsmanovic">
    <meta name="keywords" content="registracija, ISDB">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="stylesheet" href="{{asset('/')}}css/material-design-iconic-
font.min.css" type="text/css">
    <link rel="stylesheet"  type="text/css" href="{{asset('/')}}css/main-login-
reg.css">
    <meta name="_token" content="{{csrf_token()}}"/>
    <title>@yield('title')</title>
</head>
<body>
```

left-side.blade.php



```
<div id="container">
    <div id="left-side">
        <ul>
```

```
            @foreach($kategorije as $kategorija)
                <li><a href="{{route('category', ["cat" => $kategorija->putanja]
)}}"><h3>{{$kategorija->naziv}}</h3></a></li>
            @endforeach
        </ul>
    </div>
```

right-side.blade.php

```
<div id="right-side">
    @foreach($sponzori as $sponzor)
    <div class="sponsor">
        <a target="_blank" href="{{$sponzor->link}}">
            <img src="{{asset("/").$sponzor->slika}}" alt="{{substr($sponzor-
>link,7)}} " title="{{substr($sponzor->link,7)}}" width="300" height="150">
        </a>
    </div>
    @endforeach
</div>
</div>
```

--------------------------------------Requests--

AdminInsertSeriesRequest.php

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class AdminInsertSeriesRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
```

```
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
                'name' => 'required|min:5|max:60',
                'text' => 'required|min:20',
                'year' => 'required|regex:/^[\d]{4}$/',
                "img" => "required|file|mimes:jpg,jpeg,gif,png|max:2000"
        ];
    }
    public function messages()
    {
        return [

            'name.regex' => "Ime mora imati od 5 do 60 karaktera!",
            'text.regex' => 'Tekst mora imati najmnaje 20 karaktera!',
            'year.regex' => 'Godina mora imati četiri cifre!',
            "img.mimes" => "Nije dozvoljena ekstenzija!",
            "img.max" => "Maksimalna veličina slike je 2 MB!",
            "required" => "Slika nije odabrana!"

        ];
    }
}
```

AdminInsertUserRequest.php

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class AdminInsertUserRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            "username" => "required|regex:/^[a-zšđčćž0-9]{6,15}$/|unique:korisnik,username",
            "password" => "required|regex:/^[A-ZŠĐČĆŽa-zšđčćž?!&#$%@0-9]{8,15}$/",
            "firstname" => 'required|regex:/^([A-ZŠĐČĆŽ][a-zšđčćž]{2,15})(\s[A-ZŠĐČĆŽ][a-zšđčćž-]{2,15})*$/',
            "lastname" => 'required|regex:/^([A-ZŠĐČĆŽ][a-zšđčćž]{2,25})(\s[A-ZŠĐČĆŽ][a-zšđčćž-]{2,25})*$/',
            "email" => 'required|email',
            "passwordConfirm" => "same:password"
        ];
    }
    public function messages()
    {
        return [
          "username.unique" => "Korisnicko ime je zauzeto!"
        ];
    }
}
```

FeedbackRequest.php

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class FeedbackRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
```

```php
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            "ime" => "regex:/^[A-ZŠĐČĆŽa-zšđžčć\s]{2,25}$/",
            "prezime" => "regex:/^[A-ZŠĐČĆŽa-zšđžčć\s]{2,25}$/",
            "email" => "email"
        ];
    }
    public function messages()
    {
       return [
         "ime.regex" => "Ime nije u dobrom formatu",
         "prezime.regex" => "Prezime nije u dobrom formatu",
         "email.email" => "Email nije u dobrom formatu"
       ];
    }
}
```

GenreRequest.php

```php
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class GenreRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            "category" => "unique:link,naziv|regex:/^[A-ZŽĆČŠĐ][a-zšđčćž]{2,15}(\s[a-zšđčćž]{2,15})*$/"
        ];
    }
}
```

LogRequest.php

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class LogRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            "username" => "required|regex:/^[a-zšđčćž0-9]{6,15}$/",
            "password" => "required|regex:/^[A-ZŠĐČĆŽa-zšđčćž?!&#$%@0-9]{8,15}$/"
        ];
    }
    public function messages()
    {
        return [
            "username.regex" => "Korisničko ime ne sme biti kraće od 6 i duže od 15 karaktera!Nisu dozvoljeni specijalni karakteri i velika slova!",
            "password.regex" => "Lozinka ne sme biti kraća od 8 i duža od 15 karaktera! Dozvoljeni su specijalni karakteri i velika slova!"
        ];
    }
}
```

PictureRequest.php

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class PictureRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            "img" => "required|file|mimes:jpg,jpeg,gif,png|max:2000"
        ];
    }
    public function messages()
    {
        return [
            "img.mimes" => "Nije dozvoljena ekstenzija!",
            "img.max" => "Maksimalna veličina slike je 2 MB!",
            "required" => "Slika nije odabrana!"
        ];
    }

}
```

RegRequest.php

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class RegRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            "username" => "required|regex:/^[a-zšđčćž0-
9]{6,15}$/|unique:korisnik,username",
            "password" => "required|regex:/^[A-ZŠĐČĆŽa-zšđčćž?!&#$%@0-
9]{8,15}$/",
            "firstname" => 'required|regex:/^([A-ZŠĐČĆŽ][a-zšđčćž]{2,15})(\s[A-
ZŠĐČĆŽ][a-zšđčćž-]{2,15})*$/',
            "lastname" => 'required|regex:/^([A-ZŠĐČĆŽ][a-zšđčćž]{2,25})(\s[A-
ZŠĐČĆŽ][a-zšđčćž-]{2,25})*$/',
            "email" => 'required|email',
            "passwordConfirm" => "same:password"
        ];
    }
}
```

SendCommentRequest.php

```
<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class SendCommentRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            "comment" => "required|regex:/^[a-zšđčćž0-9A-ZŠĐČĆŽ?!\.\s]{5,200}$/"
        ];
    }
}
```

SponsorRequest.php

```
<?php
```

```
namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class SponsorRequest extends FormRequest
{
    /**
     * Determine if the user is authorized to make this request.
     *
     * @return bool
     */
    public function authorize()
    {
        return true;
    }

    /**
     * Get the validation rules that apply to the request.
     *
     * @return array
     */
    public function rules()
    {
        return [
            "link" => "url",
        ];
    }
    public function messages()
    {
        return [
            "link.url" => "Link nije u ispravnom formatu!"
        ];
    }
}
```

---------------------------------------Middleware------------------------------------

AdminMiddleware.php

```php
<?php

namespace App\Http\Middleware;
use Illuminate\Support\Facades\Log;
use Closure;

class AdminMiddleware
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure  $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        if(!$request->session()->has('user')){
            Log::alert("Neautorizovani korisnik sa ip adresom: ".$request->ip(). "je pokusao da pristupi admin panelu");
            return redirect()->route("pocetna");
        }
        $user = $request->session()->get('user');
        $username = $request->session()->get('user')->username;
        if($user->ID_uloga!==1) {
            Log::critical("Korisnik sa korisnickim imenom" . $username . "pokusao je da pristupi admin panelu");
            return redirect()->route("pocetna");
        }
        return $next($request);
    }
}
```

AuthMiddleware.php

```php
<?php

namespace App\Http\Middleware;

use Closure;

```

```php
class AuthMiddleware
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \Closure  $next
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        if($request->session()->has('user')){
            return $next($request);
        }
        return redirect()->route('autor');
    }
}
```